A KNOWLEDGE-BASED EXPERT SYSTEM CONSULTANT FOR OPTIMUM STRUCTURAL SYNTHESIS

By

JAHAU LEWIS CHEN

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN
PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1987

ACKNOWLEDGMENTS

The author wishes to express his sincere appreciation to his supervisory committee chairman, Dr. P. Hajela, for his invaluable guidance, support and encouragement during every stage of this study.

He would like to thank Dr. G. E. Nevill, Jr., Dr. M. A. Eisenberg, Dr. C. C. Hsu, and Dr. M. I. Hoit for serving on his committee.

Finally, the author extends his deepest appreciation to his wife, Melody, and his parents for the years of patience, encouragement and love they have provided.

TABLE OF CONTENTS

Abstract of Dissertation Presented to the Graduate School of the University of Florida in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

A KNOWLEDGE-BASED EXPERT SYSTEM CONSULTANT FOR OPTIMUM STRUCTURAL SYNTHESIS

By

JAHAU LEWIS CHEN

August, 1987

Chairman: Dr. Prabhat Hajela
Major Department: Engineering Sciences

This study brings the techniques of artificial intelligence (AI) into the optimum structural design domain. A knowledge-based expert system approach for optimum structural synthesis is presented. This approach results in the accumulation of a body of heuristic knowledge about the optimum structural design domain into a computer and emulates the reasoning processing of a human expert to provide knowledgeable advice about the difficult task of optimum structural synthesis.

An expert system OPSYN (OPtimum SYNthesis) that gives a designer interactive assistance in the computer-aided optimum design of structural and mechanical systems is developed. The knowledge base for this system includes rules for facilitating finite element modeling, optimum design modeling, and selection of optimization strategies

and parameters. OPSYN represents knowledge in IF-THEN rule form and separates the knowledge base into different knowledge modules. The environment for this development is an inference engine, called INFER, with both forward and backward reasoning capabilities, a detailed explanation facility, and an automated knowledge-acquisition system with a knowledge-base editor facility.

The use of a computer-aided design (CAD) interface with its significant data structure and graphics capabilities in various aspects of expert system development, such as knowledge acquisition and knowledge representation, is demonstrated with the OPSYN expert system.

CHAPTER I

INTRODUCTION

Motivation

During the past two decades, the development of finite element analysis techniques [1-2] and mathematical programming methods [3-4] has provided the engineer with a useful design tool by using the computer to provide solutions to engineering design problems. In recent years, the application of these techniques to structural optimization problems has approached a level of maturity that justifies the inclusion of optimum structural synthesis into the computer-aided design (CAD) environment [5-7]. This is clearly indicated by the optimization software that is included in some commercial finite element programs [8-9], and by the application of these techniques in both the automotive [10] and aerospace industries [11].

However, there has not been a significant utilization by the design community. Most realistic problems encountered in structural design have a large number of design variables and constraints. Application to these problems has been limited by the computational cost and by the complexities of performing the structural optimization process. Several useful techniques [12] have been proposed

to improve the efficiency of the process. The proper use of these techniques can make the optimization of realistic large-scale structures more amenable [13].

At the present time, the role of the computer in CAD environments is restricted to algorithmic analysis and graphics. However, many problems associated with the optimum structural synthesis not only deal with algorithmic analysis and graphics, but also require the designer's judgement and experience. The current CAD environment lacks the capability to provide knowledgeable advice for the designer in performing optimum structural synthesis.

One of the tasks that the designer needs extensive knowledge to perform is that of finite element modeling. Many interactive graphics-based automated mesh generation programs [14] are available to reduce the labor in finite element modeling. However, modeling is not only restricted to mesh generation. Without detailed knowledge of finite element modeling and the capability of each element in different programs, mismodeling can occur very frequently [15].

Similar problems exist in the formulation of an optimum design model and in the performance of optimization algorithms. The quality of the design model and the choice of a suitable optimization algorithm with its proper parameters will affect the results of structural optimization. It is recognized that optimum design modeling and the selection of a suitable optimization algorithm

requires knowledge about the specific problem and the performance of each optimization technique. Such insight is, in general, available to a few experts in the field and there is a need to disseminate this information to a larger user community.

In order to overcome those weaknesses, there is an urgent need to have the optimum structural synthesis expertise available in the CAD environment. The rapidly developing technology of knowledge-based expert systems [16], the applications of artificial intelligence (AI) techniques [17], provides a useful solution to this problem.

The present research was initiated with the aim of developing a knowledge-based expert system OPSYN (OPtimum SYNthesis) that is primarily intended as an online consultant to provide interactive assistance in the computer-aided optimum design of mechanical and structural systems. Efficient methods for knowledge acquisition, knowledge representation, and inference-reasoning strategies used in building the OPSYN expert system are also studied.

Literature Review

Several publications have explored the ideas of applying knowledge-based expert systems to finite element modeling [18-20]. However, few efforts have been made to actually develop a knowledge-based expert system for this clearly formidable task.

The first system dealing with the finite element method is SACON [21]. SACON was developed to recommend the analysis procedure for the finite element program MARC. Another system, developed by Rivlin, Hsu, and Marcal [22], performs the same task as SACON. However, both systems lack the capability of assisting the designer in generating the finite element model.

The recently developed ESSDAN expert system [23] contains some rules for modeling for a finite element program, FIPTIF, which is used for stress analysis in strip drawing processes.

The ADEPT expert system [24] integrates a CAD solid modeling system with an expert system which contains finite element modeling knowledge. ADEPT uses a combination of features deduced from the structure's geometry and information supplied by the designer to determine a modeling strategy and to automatically generate input files for finite element analysis programs. Currently, ADEPT can give the designer advice only in the task of element selection.

Some attempts on the application of artificial intelligence techniques in the field of optimum design are documented in References 25, 26, 27, and 28. In those papers, the authors explored the ideas of using engineering knowledge and artificial intelligence tools to improve the optimization performance.

Li and Papalambros [29] developed a production system by using monotonicity analysis to improve optimization performance.

Recently, the EXADS expert system [30] has been developed to advise the designer in the selection of an optimization algorithm from the ADS general-purpose optimization program [31].

An artificially intelligent nonlinear optimization expert system IDEAS [32] is being developed by Baenziger and Arora. IDEAS is attempting to become an optimization expert to assist the user in performing nonlinear optimization algorithms.

Hartmann [33] applied the symbolic manipulations technique for reanalysis problems in structural optimization. A rule base for symbolic differentiation of the finite element matrices with respect to the design variables was developed to aid the optimization process.

Organization

The remainder of this dissertation is divided into seven chapters. The second chapter reviews the tasks involved in the optimum structural synthesis. Those tasks that need extensive engineering knowledge to perform are identified.

In Chapter 3, the knowledge-based expert system approach is presented. The tasks in each component of the expert system are introduced. The tools available for the development of an expert system are also discussed.

Chapter 4 presents an overview of the OPSYN expert system. The architecture of this system is presented and

the organization of knowledge base is outlined. The knowledge-representation scheme used in this system is introduced with examples. An automated knowledge-acquisition system embedded in a CAD environment is also presented.

The INFER inference engine that provides the environment for this development is described in Chapter 5. Details of the inference-reasoning techniques used in INFER are presented. The knowledge-base editor facility to aid the knowledge engineer in developing the knowledge base is introduced. Some special functions for coupling the expert system with conventional programs are also discussed.

A detailed description of the knowledge in each knowledge base module is contained in Chapter 6.

Illustrative examples are presented in Chapter 7 to demonstrate the use of the OPSYN expert system.

Finally, conclusions drawn from this study and recommendations for future research are given in the final chapter.

CHAPTER II

OVERVIEW OF OPTIMUM STRUCTURAL SYNTHESIS

Introduction

The earliest analytical works in the field of optimum structural synthesis are represented in a 1869 publication of Maxwell [34] and a 1904 publication of Michell [35]. Although these works had restricted practical applicability, they offered valuable insight into the subject of optimum structural synthesis. During the early and mid forties, the "simultaneous failure mode theory" was developed for structural component optimization. On the basis of this theory, the structure was sized in such a manner that it failed simultaneously in all of its failure modes. In the early fifties, linear programming techniques were used to obtain the optimum design of frames based on the plastic collapse design philosophy.

Since 1960, optimum structural synthesis has developed in two directions. The first approach is the use of nonlinear programming algorithms in conjunction with finite element and other discrete analysis methods. Another approach is the use of optimality criteria methods [36]. The latter approach is based on the derivation and satisfaction of a criterion that defines the optimum for a

structure for given loading conditions and prescribed constraints. The satisfaction of this optimality criterion leads, iteratively, to an optimum design. This approach is limited in that the optimality criterion is problem-dependent and often difficult to derive.

The use of nonlinear mathematical programming techniques in optimum structural synthesis was first introduced by Schmit [37] in 1960. It provides a more direct and flexible design procedure for structural optimization. These techniques have being broadly applied and recognized as a useful design tool. Today, the state-of-the-art in optimum structural synthesis has matured from a concept to a practical design tool. Details of this development can be found in Reference 38.

As illustrated in Figure 2-1, optimum structural synthesis may be viewed as a sequence of steps that must be performed to obtain an optimum design. It involves the generation of a finite element model for analysis, generating an efficient optimum design model, finite element analysis, and use of an optimization strategy in conjunction with the analysis. There is a sequence of iterations between the finite element analysis and the optimization algorithm, until a converged solution is obtained. Details of optimum structural synthesis methods can be obtained from Reference 39. A brief overview of the principal tasks in optimum structural design, which need extensive knowledge and understanding, is provided in the following sections.

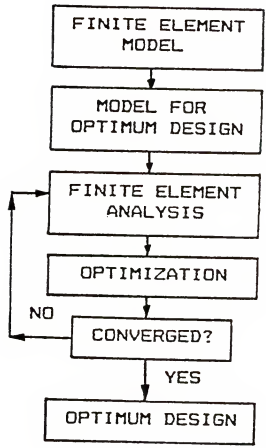


Figure 2-1. The flowchart of optimum structural synthesis.

Finite Element Modeling

Finite element analysis is a numerical technique for obtaining approximate solutions to engineering problems. This approach solves the problem by using a finite element model, which is composed of a number of nodes and elements, to approximate the real problem domain, as shown in Figure 2-2. Generating the finite element model usually involves idealization and discretization.

The proper idealization of the actual problem is the most important step in finite element modeling, and refers to selecting the types of elements. The element's type is characterized by the properties (i.e., truss, beam, membrane, plate, etc), orders (i.e., linear, quadratic), or shapes (i.e., triangular, quadrilateral). The selection of appropriate elements for the problem at hand requires knowledge of the characteristics of each element available within the library, and of the physical problem. This includes recognition of the failure modes one may expect under a prescribed load, the presence of loads, geometry of the domain, and the capabilities and limitations of each element. Unfortunately, no general guidelines for choosing the appropriate element can be obtained, since the type of element that yields good accuracy with low computing time is problem-dependent.

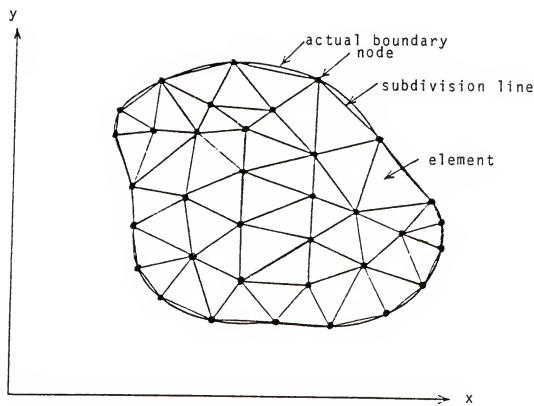


Figure 2-2. The finite element model of actual problem domain.

In addition, there is a growing concern about the reliability of commercial finite element codes. Recently, an agency in the U.K. [40] was organized to propose a set of test-bed problems to standardize the performance of finite element codes. The initial results of this effort have been published recently [41]. It is expected that these and similar test results will give the designer some information in selecting appropriate elements.

The discretization of the structural or mechanical system involves decisions on the number, shape, size, arrangement, and distribution of finite elements in the problem domain. Proper specification of node locations and element shapes is essential in order that the discretized domain and the actual domain are as close to each other as possible. Using many simple elements to model the domain may be equivalent to using a few high-order elements. For example, a curved boundary may be represented by a few elements with curved boundaries (isoparametric elements) or a larger number of straight-edge elements.

Due consideration must also be given to an accurate representation of concentrated loads, distributed loads with and without discontinuities, and material and geometric discontinuities, including cutouts and reentrant corners. If possible, one should try to align the subdivision lines with the boundary of structure domain and with principal loading trajectories.

Another consideration is to discretize the domain or portions of the domain into sufficiently small elements to resolve the spatial variations in the solution. This requires an a priori understanding of the solution and an estimate of the computational costs involved. Typical rules-of-thumb require that one consider element aspect ratio, element distortion, any symmetry or lack thereof of loading and geometry in the problem domain, and the node numbering sequence.

The mesh should be devised in such a way that abrupt changes in element size are minimized. There has been an increased focus on automated mesh generation and mesh refinement techniques [42] to improve the mesh layout. However, these techniques are not always practically feasible, and one must often depend on one's own judgement for this task.

Optimum Design Modeling

Structural optimization is a numerical design technique for obtaining optimum design of complex practical structures. The mathematical formulation of a structural optimization problem can be stated as follows:

Minimize $W(d)$

Subject to $g_j(d) \leq 0$ $j= 1,2,...,m$

$h_k(d) = 0$ $k= 1,2,...,q$

$d_i^l \leq d_i \leq d_i^u$ $i= 1,2,...,n$

where W is the objective function; d is a vector of design variables, each component of which is limited by lower and upper bounds, d_i^l and d_i^u, respectively; g_j and h_k are inequality and equality constraints on the response quantities, respectively.

Realistic structures may have hundreds of design variables and thousands of implicit nonlinear constraints. To improve the efficiency of optimization performance, several approximate techniques have been developed. These include reduction in the number of independent design variables by design variable linking, reduction of constraints by using efficient constraint representation, limiting the number of repeated finite element analyses through approximation concepts, and by decomposing the problem into a sequence of smaller problems [43].

Design variable linking schemes [13] are invoked to reduce the number of independent design variables. Variables are grouped into sets with one member of the set acting as the design variable in the optimization problem. All other variables in the set either are similar or have a predetermined relationship to this representative variable. For example, the pertinent cross-sectional properties of a finite element may be subjected to change during the optimization sequence. In the event of a very large number of finite elements for the structure, not all of the cross-sectional properties are allowed to vary independently. Manufacturing considerations may not allow this to be a physically amenable choice in most practical structures. Another scheme of design variable linking is the controlled growth method [44] that reduces the number of design variables by linking the growth of a single dominant variable to others in the active set.

Efficient methods of constraint representation include the process of constraint deletion [13], where the inactive constraints are temporarily ignored during the optimization process. The cumulative constraint formulation [44,45] is another in a class of efficient constraint representation schemes that replaces multiple inequality constraints by a single cumulative measure.

There are three basic techniques of approximation analysis [13], namely, Taylor series approximation, reduced basis approximation, and subspace iteration methods. First

order Taylor explicit approximation is the most commonly used approach and requires that the functions to vary linearly with design variables, or are, at worse, mildly nonlinear, to obtain accurate approximations. For optimization problems dealing with vibration analysis or dynamic loading, the reduced basis approximation and subspace iteration method can be used for structural reanalysis. Details of these two methods can be obtained in Reference 46.

The multilevel decomposition approach divides a large structural optimization problem into a hierarchy of much smaller subproblems, and optimizes each of the subproblems separately. Equality constraints are placed at every level to ensure consistency between subproblems or levels. A large structural optimization problem may be decomposed into small subproblems according to different disciplines, several substructures, or special classes of problems such as composite sandwich-panels or beam/frame problems. For example, an airframe structure can be decomposed into the wing structure and fuselage structure.

At the beginning of optimum design modeling, the designer must consider whether it is necessary to establish an efficient design model for large-scale structural problems. The finite element model may be used to generate the optimum design model. However, for realistic large-scale structures, the finite element model is too large and is not adequate for design analysis. The

possibility of generating a simplified analysis model for optimum design [47], dimensionality reduction by invoking efficient approximation techniques, or using decomposition techniques must all be considered.

The second step in formulating the optimum design model includes deciding on the kind, number, and distribution of design variables, identifying the load conditions and constraints to be considered during the optimization, and selection of a suitable objective function for the design problem at hand.

The typical design variables are the cross-sectional properties of structural elements (i.e., cross-sectional area, moment of inertia, thickness, torsional rigidity, etc.), the number and mutual connectivity of members and joints in a structural system, the coordinates of joints in a structural system, material properties, orientation of each layer in composite material, and the shape of external boundaries and interfaces of a structure.

The constraints may be the restrictions imposed on design variables, the upper or lower bounds on design variables, the state equations governing the structural response associated with the loading conditions, or the specific restrictions on the response of the structure (i.e., stresses, displacements, buckling loads, natural vibration frequencies, etc.).

Most structural optimization problems involve weight as an objective function to reflect the material cost. Other

measures of the structural performance (like displacement, stress, buckling load, natural vibration frequency, or stiffness, etc.) may be taken to represent the objective function. For optimization problems that have more than two important criteria, the objective function may be represented as a weighted sum of those functions. The choice of the weighting factor will affect the results of optimum design.

As the approximation concepts are applied to large-scale structural problems, the choice of proper intermediate design variables and constraint representation is very important to the quality of the approximation. For example, in designing truss structures, the basic design variables are the cross-sectional area d=A. If the intermediate design variable d=1/A is selected, then the first order Taylor series approximation to the displacement with respect to this intermediate design variable is precise for statically determinate truss structures. For mildly indeterminate truss structures, it would improve the accuracy of the linear assumption for displacement constraints. Similarly, in plate bending structures, the intermediate variable would be the inverse of cubic of thickness $d=1/t^3$. It is obvious that the choice of proper intermediate design variables is problem-dependent.

Furthermore, after the design model is formulated, the designer also needs to determine a suitable optimization strategy and sensitivity analysis method for efficient

optimization performance. The available strategies are fully stressed design [48], geometric programming [49], dual methods [50], optimality criteria method, and mathematical programming. Each strategy offers certain advantages for some types of optimization problems. For example, in a structural design problem that is constrained only by allowable stress requirements, the fully stressed design method is a suitable choice. Geometric programming is a good choice for problems where the objective function and constraints are, or can be represented in, a generalized positive polynomial form. Dual methods have the advantage of dealing with discrete design variable problems. The selection of these strategies is based on the properties of the optimum design model.

Sensitivity analysis deals with the computation of derivatives of response measures with respect to design variables. Design sensitivity analysis methods include finite difference method [51], design space method, behavior space method, and the virtual load technique [52]. Since most of the optimization algorithms require the sensitivity information and the calculation of sensitivity analysis is computationally demanding, the choice of a suitable method becomes very important. The choice of these methods is dependent on the characteristics of the design model.

Optimization Performance

The selection of the best optimization algorithm for the problem under consideration is the first step in enhancing the optimization performance. The first requirement of a good algorithm for engineering optimum design is the reliability of the algorithm, where reliability refers to the ability to locate at least a local optimum point, regardless of the starting points. Another requirement is the efficiency of the algorithm such as the computer run-time involved in obtaining a solution to the optimization problem. However, no single algorithm is available that really achieves both requirements [53].

The available optimization algorithms can be divided into unconstrained optimization and constrained optimization techniques. The latter can be further separated into direct methods which solve the constrained problem directly and indirect methods which change the constrained problem to an unconstrained one. Each method has its own advantages and drawbacks. The detailed mathematical theory of each algorithm can be obtained in References 3 and 4. The available mathematical programming optimization techniques and their classifications are illustrated in Figure 2-3.

One of the reasons for the existence of several algorithms is that no single algorithm can work well under all conditions. This fact has been verified in a recent study [53]. Since no one optimization algorithm can

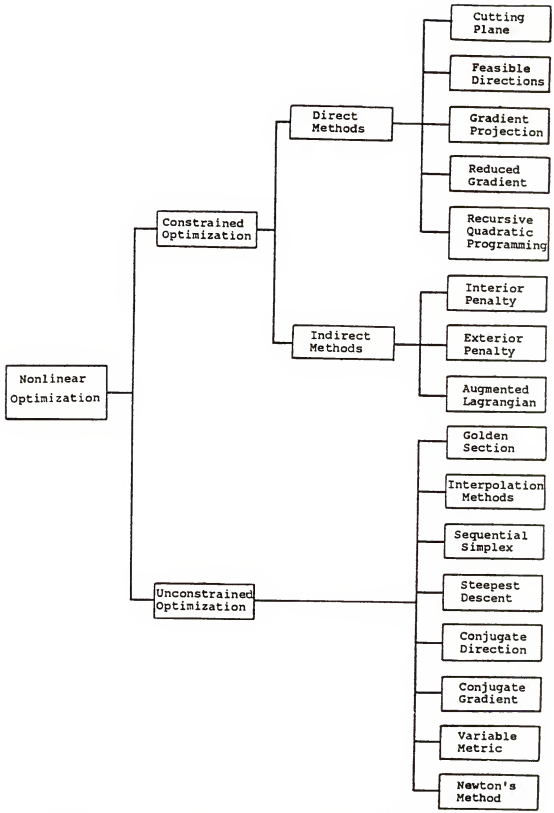


Figure 2-3. Methods of mathematical programming.

efficiently solve all classes of optimization problems, the designer must understand the capability of each optimization algorithm and the characteristics of the optimization design model in order to make the appropriate choice. However, structural design engineers are not familiar with the intricacies of these optimization techniques. Although there are several comprehensive studies [53-55] that evaluate various methods and algorithms, the task of choosing the best algorithm is still difficult for an inexperienced designer.

After choosing the best algorithm, the designer also needs to select the user-defined program parameters (for example, convergence criterion, active constraint strategy, push-off factors, step size, etc.) before actually executing the optimization algorithm. The values of those parameters can significantly influence the optimization performance. The choice of those parameters is dependent upon the behavior of the algorithm and the characteristics of the design model. An experienced designer has compiled knowledge accumulated from a wide variety of design problems. A newcomer to the field may go through an extended period of time, learning about the possible pitfalls and anomalies in the use of such methods.

During the iterative optimization process, the designer may be required to diagnose the difficulties in optimization performance and to modify the algorithm parameters or the original design model itself. Sometimes, the designer may

need to choose another starting point for the optimization process or switch to another algorithm to achieve optimum performance.

Recently, there has been an increased focus on the use of the hybrid approach [56], which employs several different optimization algorithms for efficient and reliable optimization performance. General-purpose optimization programs, such as ADS [31], provide the designer with an option of combining different algorithms to perform optimization efficiently.

The foregoing discussion indicates that a considerable amount of experience and engineering judgement is needed for performing optimum structural synthesis. This situation provides an excellent opportunity for application of knowledge-based expert system techniques. The knowledge-based expert system approach will be explained in the next chapter.

CHAPTER III

KNOWLEDGE-BASED EXPERT SYSTEM APPROACH

Introduction

Knowledge-based expert system technology has been one of the most rapidly developing branches of artificial intelligence (AI) in the past few years. Artificial intelligence is the science of creating intelligent behavior on the computer. Since intelligent reasoning requires knowledge, one of the major subjects in AI is the study of knowledge and utilizing knowledge.

Expert systems are computer programs that capture specialized knowledge about a narrow and well defined domain and can simulate the reasoning process of a human expert to provide knowledgeable advice about a difficult task. The domain for expert system development must be a narrow and well defined. Furthermore, the task itself must contain some degrees of difficulty that only a few experts can resolve in a satisfactory manner. Past experiences indicate that if the domain is not adequately focussed, attempts at configuring expert systems will not be very successful.

The difference between expert systems and conventional programs are illustrated in Figure 3-1. The conventional programs store data in a data file or data base and use

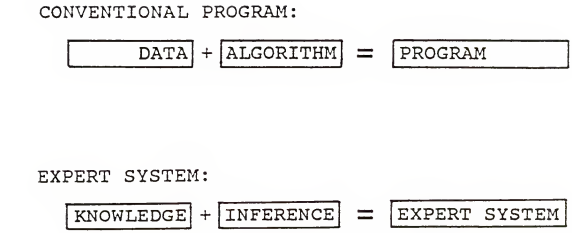


Figure 3-1. The difference between the expert systems and the conventional programs.

algorithms repeatedly to manipulate data to give numerical results. Although some knowledge may be contained in the algorithm, modification of the algorithm as the problem domain is better understood is not an easy task. For expert systems, the knowledge is stored in a knowledge base. Inference reasoning techniques are invoked to arrive at conclusions.

In this chapter, the structure of expert systems and the techniques used in developing expert systems are introduced and discussed.

Components of Expert System

Expert systems should typically consist of four components: a knowledge base, an inference engine, a knowledge-acquisition facility, and an input-output interface with an explanation facility [17], as shown schematically in Figure 3-2. Currently, most expert systems lack a good knowledge-acquisition facility. It is desirable to separate the knowledge base and the inference engine. With this separation, the knowledge can be modified as the domain is better understood. Such an approach also allows the same inference engine to deal with knowledge bases pertaining to different problem domains.

A knowledge base consists of domain facts and heuristic knowledge associated with the problem. Domain facts are the knowledge and understanding of basic principles. Heuristic

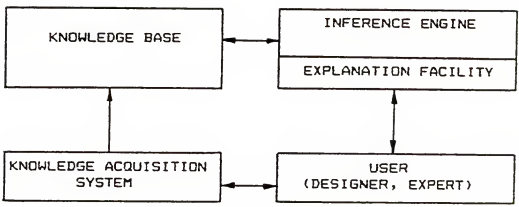


Figure 3-2. Architecture of an expert system.

knowledge is developed from the application of basic principles for a class of problems.

The inference engine, which contains the reasoning process, attempts to match knowledge input facts with the knowledge in the knowledge base, to draw conclusions and produce an explanation. Some reasoning techniques are described in a later section.

Knowledge acquisition is the process by which expert knowledge is obtained from the source of knowledge for representation in a knowledge base. This process of building expert systems is called knowledge engineering and is performed by the knowledge engineer.

The input-output interface provides an interface between the user and the expert systems. It allows the user to create or modify a knowledge base by a knowledge-base editor facility, and use of this knowledge base with an explanation facility for solving the problem at hand.

The choice of a suitable knowledge-representation scheme and reasoning technique is an important ingredient to the success of an expert system. During the development of an expert system, the primary bottleneck is the acquisition of the expert knowledge for the knowledge base. A detailed description of expert-systems concepts and expert-system-development technology is available in References 57 and 58. A brief review of each component and the associated techniques is presented next.

Knowledge Representation

In developing a knowledge base, an important issue confronting the engineer is how the knowledge is to be represented in the knowledge base. There are many knowledge-representation schemes, any of which can be used alone or in conjunction with others to represent knowledge. Each scheme has certain advantages. A detailed summary of such schemes is found in Reference 59. In this section, the three most widely used knowledge-representation schemes in current expert systems, rule-based, frame-based, and logic-based, are described.

In the representation of knowledge within the knowledge base, the rule-based (also referred to as production rules) knowledge representation is the most popular scheme in current expert systems. In this approach, the knowledge is represented as a series of IF-(condition)-THEN-(action) statements. For example,

IF

This problem is beam or frame

AND

If this problem has more than one plane of bending and any combination of axial, bending and torsional loads,

THEN

You should choose E21 elements for this problem. The E21 are general straight or circularly curved beam elements in EAL finite element program, such as channels, wide-flanges, angles, tubes, zees.

The (action) part is executed if the (condition) part provides a match with the available facts. The rules which contain the knowledge of using other rules are called meta-rules [60].

The rule-based scheme provides a natural way for describing complex knowledge. The use of the rule-based scheme also simplifies the job of explaining how the expert systems reached a certain conclusion. One drawback of using this scheme is that it may introduce contradictions as new rules are added or existing rules are modified.

Another scheme is the frame-based knowledge representation. This is a network data structure to represent the relations between concepts, objects, or events, and their attributes. This scheme allows more relations among the facts. Semantic networks and frames are included in this scheme, as shown in Figures 3-3 and 3-4, respectively.

Semantic networks use nodes to represent concepts, objects, or events, and arcs describing the relations between the nodes. Arcs can be defined according to the type of knowledge being represented. For example, the statement "Beam element is an one-dimensional element" can be represented in a simple semantic net by using the is-a relation, as show in Figure 3-3. This is-a relation allows the lower nodes in the net to inherit information from the higher nodes in the net. This property is referred to as inheritance.

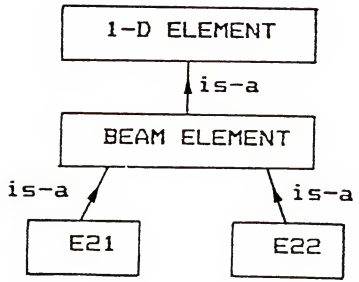


Figure 3-3. Semantic network knowledge representation.

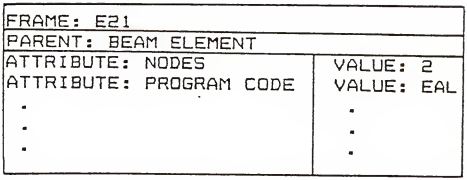


Figure 3-4. Frame knowledge representation.

A frame is composed of a node and a set of slots that together describe an object, concept, or event. Each slot may be an attribute of the node or a link to another node. The latter provides the frames with the inheritance properties. The slot may also have procedures associated with it. The procedure contains a computer code to monitor the information to the node when the value of the slot is changed. An example of frames is shown in Figure 3-4. The name of this frame is "E21." Two slots, "nodes" and "program code," have values "2" and "EAL," respectively.

The logic-based scheme [61] uses first order predicate logic [62], a special subset of mathematical logic, to represent knowledge. PROLOG [63] is one of the more commonly used logic programming languages. The key idea of this scheme is programming the knowledge to describe the known facts and relationships in the problem domain rather than to prescribe the sequence of steps taken by the computer to solve the problem. For instance, a fact statement "Beam element is an one-dimensional element" can be represented in PROLOG as

is a (beam element, one-dimensional element).

The above description is called predication. The basic units for representing facts or rules are predications that are presented by a predicate name (i.e., "is a") followed by a list of arguments. The rule is represented in a set of

clauses. Each clause has the following form:

consequent : [antecedent-1, antecedent-2,...antecedent-n].

The antecedents and consequent in each clause are predications. The consequent is true if its antecedents can be proven true. An example of a rule represented by logic-based scheme is as following:

```
is a (E21 element, one-dimensional element) :
[ is a (E21 element, beam element),
  is a (beam element, one-dimensional element) ].
```

This scheme has the feature of both representing and making inferences on knowledge by using one logical formulation.

Inference Engine

The reasoning techniques, also known as problem-solving techniques or search techniques, are the strategies used in the inference engine to control the problem-solving process. Three commonly used techniques are backward-chaining reasoning, forward-chaining reasoning, and mixed reasoning. In this section, details of those techniques are described. A more detailed description of problem-solving strategies is given in Reference 59.

The backward-chaining reasoning starts from a hypothesis or goal to check if the hypothesis can be supported by the available facts. If the known facts cannot support the hypothesis, then the needed facts are set up as

a sub-hypothesis. This approach has the drawback of perhaps becoming fixed on an initial set of hypotheses and having difficulty shifting to others as the available facts do not support them.

On the other hand, the system that works from known facts to the conclusions is called forward-chaining reasoning. This technique is suitable for the problem that has a large number of hypotheses and only few known facts. It sometimes has the disadvantage of generating many hypotheses not directly related to the problem.

The mixed reasoning technique combines both forward-chaining and backward-chaining. It uses both reasoning techniques to yield conclusions.

Many inference engines can handle incomplete knowledge by associating confidence levels with the facts. The range of confidence level can be, for example, from 1.0 (most certain) to 0.0 (least certain). Each rule is assigned a confidence level by the domain expert. The combination of this confidence level with the one which is responded to by the user about how well the rule applies to the problem is confirmed as the state of this rule. For example, if the domain expert has given a confidence level of 0.9 on a rule and the user responds to this rule with a confidence level of 0.8, then the combined confidence level is 0.9*0.8=0.72. For "and" combination of rules, the new confidence level equals the first rule's confidence level multiplied by the second rule's confidence level. In case of an "or"

combination of rules, the new confidence level is combined according to the following computation:

```
new-confidence-level = confidence-level-one
    + ((1.0 - confidence-level-one) * confidence-level-two)
```

The conclusion from the inference engine is the one which has the highest confidence level associated with it.

Knowledge Acquisition

Currently, the knowledge-acquisition process is performed by the knowledge engineer who builds the expert system. The sources of expert knowledge are the domain expert and experiences of domain experts that are documented in available literature.

The process of knowledge acquisition may be divided into five steps [58]. The first step is identification and refers to characterizing the important aspects of the problem, such as participants (knowledge engineer and domain experts), problem domain, knowledge sources, and goals. The conceptualization is the second step. During this stage, the key properties and relations are made explicit and some ideas of knowledge representations and tools are considered. The third step is the formalization process, and begins to map the model of the task and its key properties and relations into some representation schemes. The fourth step is implementation and involves mapping the knowledge into

the representation scheme associated with the tool chosen for the project. The final step is testing and revising the prototype system. Depending on the results of testing, the actions may be reformulation of the problem, redesign of the knowledge representation, or refinement of the knowledge base.

Naturally, an interview with the domain expert is one way in which a knowledge engineer can extract the knowledge from the domain expert. The difficulty of this approach is that the domain expert may not know how to describe the decision process, or may simply misunderstand the questions. It also requires the knowledge engineer to be familiar with the problem domain. To keep the problem at a manageable level, it is important to control the extent of the problem domain.

Another approach to eliciting knowledge is protocol analysis [64]. This technique avoids interviewing the domain expert directly by letting the domain expert perform specific examples of problems and recording the problem-solving process in a transcript. The knowledge engineer analyzes this transcript with the domain expert's help, if available, to elicit domain knowledge. This technique gives the domain expert more freedom in expressing his expert knowledge and serves a useful function of making documented case histories available. However, the gaps between the knowledge engineer and the domain experts still exist.

Other automated acquisition techniques [65] under development, such as the induction method [66] and repertory grid technique [64], may overcome this difficulty in the near future.

Several useful algorithms [66-67] have been developed to use the induction method to elicit the knowledge. This method needs the domain expert to provide a set of examples of different types of decisions and the attributes which influence the decision. The algorithm uses the examples to induce rules. The advantage of this method is that the domain expert does not actually have to describe the decision-making process himself.

The repertory grid technique is based on a psychological personal construct theory [68]. This theory shows that every individual has his or her own model for problems in the world. This technique is a method for investigating such a model. A successful implementation of this technique is the ETS system [69]. The domain expert inputs the conclusion items, which should be made by the proposed expert system, and its rating values into the ETS system. The system constructs a rating grid and analyzes the grid to come out with rules. This technique offers the domain expert a useful way to organize his knowledge.

Expert-System-Development Tools

In recent years, several commercial expert-system-development tools have been developed for building expert

systems. The tools are three types [57]: programming languages, system-building aids, and expert system shells.

An expert system can be built by using the programming languages such as LISP, PROLOG, C, PASCAL, or FORTRAN. LISP and PROLOG are specially designed for AI applications and have been chosen for most of the work in expert systems development. Currently, there is a trend to choose other conventional languages [70], such as C, to develop expert systems for reducing operational times.

Only a few expert-system-building aids have been developed. Most of them are knowledge-acquisition systems to assist the knowledge engineer in eliciting knowledge process, such as ETS, TIMM [71], and RULEMASTER [72].

Most commercial expert-system-development tools are expert system shells. A detailed evaluation of a variety of currently available expert system shells can be found in Reference 73. Expert system shells contain a domain-independent inference engine, an empty knowledge base, and a user-friendly interface. With these shells, the knowledge engineer can concentrate on the knowledge representation and neglect the complex inference engine strategies.

However, the selection of a suitable knowledge-representation scheme and an inference-reasoning technique plays an important role to the success of an expert system. Most expert system shells are developed from domain-dependent expert systems by taking out the original knowledge base. These shells may fit in with problems

similar to the one that deals with the original expert system. The available shells should be used carefully.

For the OPSYN expert system development, an expert system shell was selected as the framework for initial development. After the initial studying and testing, a new domain-independent rule-based inference engine, INFER, was developed as the environment for the OPSYN expert system. The reasons for this choice will be made clear in subsequent chapters. An automated knowledge-acquisition system was also developed to assist the knowledge-acquisition process. The details of the INFER environment will be presented in Chapter 5. The next chapter provides an introduction to the OPSYN expert system.

CHAPTER IV

THE OPSYN EXPERT SYSTEM

Introduction

In this chapter, the development of a new expert system OPSYN (OPtimum SYNthesis) is presented. OPSYN is a concept demonstration expert system for providing a structural design engineer with interactive assistance in various aspects of computer-aided optimum design of structural and mechanical systems.

Since the OPSYN expert system is intended to be operational in a CAD environment, the role of expert systems in this environment is examined first, and is illustrated in Figure 4-1. This figure represents, at best, a somewhat optimistic view of what current CAD configurations should offer. Most existing CAD systems offer only a data base management capability, graphics capability, and more recently, some analysis features. The inclusion of optimization methods and expert systems is a logical extension to the present level and would significantly enhance the utility of the CAD systems. Expert systems will play the role of assisting the designer in performing various tasks and dealing with algorithmic programs directly to improve their performance. On the flip side, the CAD

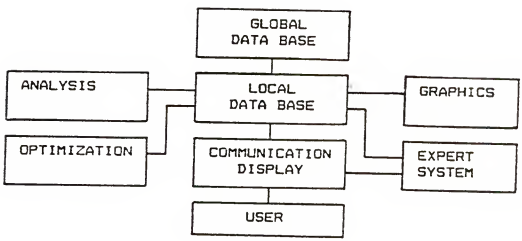


Figure 4-1. The role of expert systems in CAD environment.

environment has a well defined role in the knowledge-acquisition process. This will be presented in greater detail in a later section.

The OPSYN system consists of rules for finite element modeling, optimum design modeling, and assistance in the optimization performance for optimum design problems. The finite element program EAL (Engineering Analysis Language) [74] and the ADS (Automated Design Synthesis) general-purpose optimization program version 1.10 [75] are selected to perform finite element analysis and optimization, respectively.

At the initial stage of this development, an expert system shell AESOP (An Expert System engine with Operative Probabilities) [30] was selected as the environment for the OPSYN expert system. The AESOP uses the rule-based scheme and the backward-chaining reasoning technique for knowledge representation and inference reasoning, respectively. It allows the designer to respond with confidence levels, and also provides a help and explanation facility. AESOP is written in IQ-LISP and operates on an IBM PC-XT personal computer. After the preliminary development and testing, some limitations in AESOP were found. These included a lack of a capability that would allow the knowledge base to be linked with existing algorithmic programs, a graphic-display facility for knowledge representation to prevent misrepresentation of knowledge, an automated knowledge-acquisition system, and a user-friendly interface for

knowledge base editing and modification. Therefore, a new inference engine environment, INFER, was built for providing the environment for the OPSYN expert system development. The current structure of the OPSYN expert system is presented in the next section.

Architecture of the OPSYN Expert System

The conceptual relationship between the OPSYN expert system and the components of an optimum structural synthesis procedure is illustrated in Figure 4-2. The left hand side of this figure represents the sequence of steps that must be performed to obtain an optimum design. As discussed in the second chapter, this includes finite element modeling, optimum design modeling, and use of an optimization strategy in conjunction with the analysis. The aim of the OPSYN expert system is to integrate all three tasks into an online consultant for optimum structural synthesis. The knowledge base for each of these tasks is developed in a modular form and embedded into the INFER inference engine environment, as shown schematically in the right hand side of Figure 4-2.

The current structure of the OPSYN expert system is illustrated in Figure 4-3. It contains a knowledge base, an inference engine with an explanation facility, a knowledge-acquisition facility, and a user-friendly input-output facility which contains a knowledge-base editor and a graphical display capability. The OPSYN system is written in FORTRAN-77 language and is presently operational on a VAX

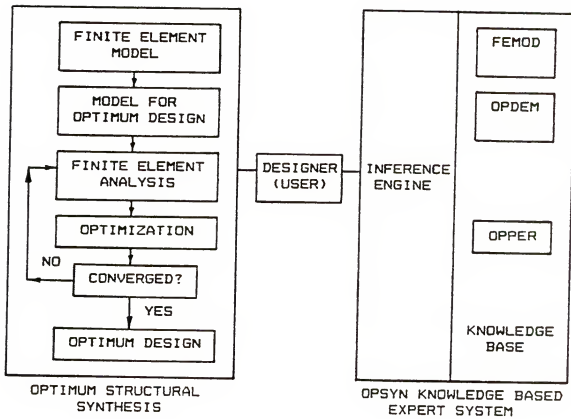


Figure 4-2. Expert system organization in the framework of optimum structural synthesis.

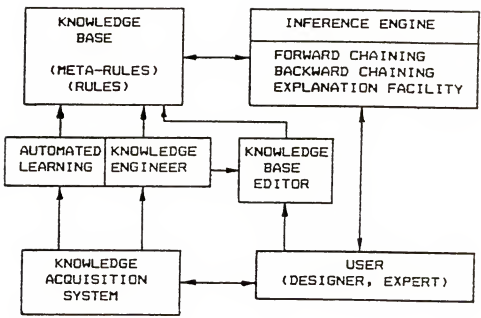


Figure 4-3. Architecture of the OPSYN expert system.

11-750 system. A Tektronix 4107 graphics terminal serves as the user interface.

The OPSYN expert system is built to allow the user (i.e., designer, domain expert, or knowledge engineer) to deal with each component through the same interface. The knowledge-acquisition facility contains two programs "ACQU2" and "RULE" for eliciting knowledge and rule evaluation, respectively. The knowledge-base editor program is called KBSED. In order to execute the OPSYN expert system, a user needs to assign the files in knowledge base to data files "INPUT" and "INPUTKB" for INFER and KBSED programs, respectively. This procedure is necessary to prevent an improper response from destroying the information in knowledge base.

Details of the INFER inference engine and its environment as well as the extent of knowledge base will be presented in Chapters 5 and 6, respectively. The other sections of the present chapter will introduce the organization of knowledge base, knowledge-representation scheme, and the knowledge-acquisition facility.

Organization of Knowledge Base

The current organization of the knowledge base for optimum structural synthesis is shown in Figure 4-4. The knowledge base is divided into three modules according to the task in different domains. The FEMOD (Finite Element MODeling) module contains the rules for finite element

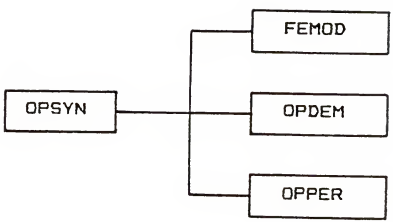


Figure 4-4. Knowledge base organization for the OPSYN expert system.

modeling. The rules for optimum design modeling and optimization performance are grouped into the OPDEM (OPtimum DEsign Modeling) module and the OPPER (OPtimization PERformance) module, respectively.

The organization of the finite element modeling knowledge base module is illustrated in Figure 4-5. In its present form, the system has the ability to assist in modeling one- and two-dimensional domains. As shown in the figure, the knowledge base itself is divided into seven files. The FEMOD file contains the meta-rules to choose other files within the knowledge base module and to recommend an inference reasoning technique. The rules for one- and two-dimensional mesh generation are available in files MESH1 and MESH2, respectively. Rules for selecting nodes and subdivision lines are stored in file NODE. Element-selection strategies are available in file ELEMENT. The SHAPE file contains rules for checking distortion, aspect ratio, and node numbering. Rules for refining the mesh are stored in file REMESH.

As shown in Figure 4-6, the knowledge base for optimum structural design modeling is itself divided into ten separate files. The OPDEM file contains the meta-rules to choose other files within the knowledge base and to recommend a suitable inference-reasoning technique. The rules for efficient model generation are stored in file APPMOD. The DECOMP file contains rules to decompose the structural optimization problem into a sequence of smaller

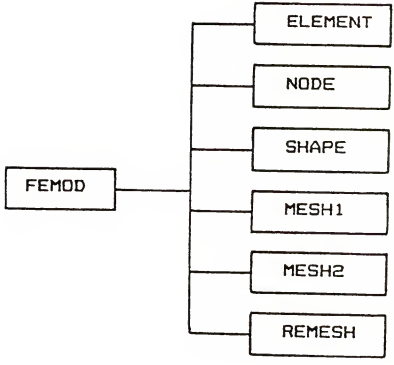


Figure 4-5. Knowledge base organization for finite element modeling.

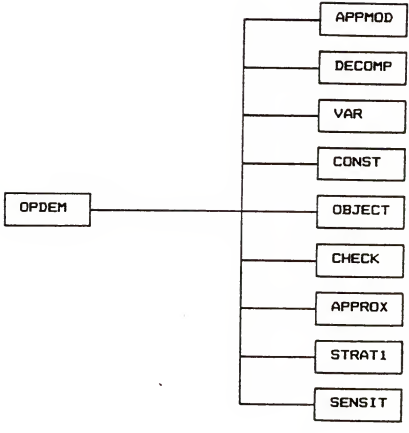


Figure 4-6. Knowledge base organization for optimum design modeling.

problems. The rules for selection of design variables, constraints, approximation techniques, strategies, and sensitivity analysis methods are available in files VAR, CONST, APPROX, STRAT1, and SENSIT, respectively. The OBJECT file contains rules to define an appropriate objective function. The rules for checking design model formulation are available in file CHECK.

As shown in Figure 4-7, the knowledge base for improving optimization performance is divided into nine files. Currently, OPPER knowledge base module has only the capability to assist the designer in choosing optimization algorithms. OPPER file contains the meta-rules to choose other files within this knowledge base module and to recommend an inference-reasoning technique. The file METHOD contains rules for choosing methods for constrained problems treated as unconstrained problems. The rules for selecting a suitable strategy for constrained problems and for constrained problems treated as unconstrained problems (by penalty function methods) are stored in files CSTRAT and CUSTRAT, respectively. The files COPT and CUOPT contain rules for selecting an optimizer for constrained problems, and for constrained problems treated as unconstrained problems, respectively. The rules for one-dimensional search method selection for both of these types of problems are available in CSEARCH and CUSEARCH, respectively. RESTART file contains rules for switching to another algorithm when the original one failed to reach a solution.

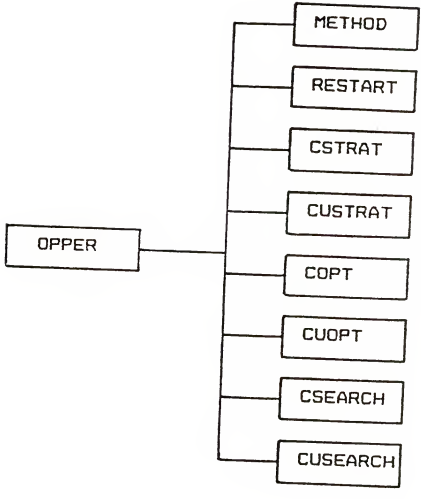


Figure 4-7. Knowledge base organization for optimization performance.

This arrangement of collapsing problem-specific rules into different files was necessary for applying suitable inference-reasoning techniques and to speed up the inference process.

Knowledge-Representation Scheme

The knowledge-representation scheme used in the current knowledge base is an IF-THEN rule-based scheme. A value of confidence level is associated with each rule. The rules can be divided into two categories, meta-rules and rules.

An example of a meta-rule in the OPDEM knowledge base is as follows:

```
THIS IS RULE 1

CONFIDENCE LEVEL = 1.0

IF

    This problem is at the first step of optimum design
    modeling

THEN

    You need to consider if the computational time and/or
    accuracy are important for this problem.  You should
    run the APPMOD knowledge base by using backward-
    chaining reasoning.
AND
    ***COPY APPMOD.KBS;1 INPUT.DAT;3
```

The "***" symbol is a function symbol in the INFER inference engine. It can create a command file which essentially contains the statement following the symbol in the rule, and execute this command file. In this example, the command file will copy the APPMOD knowledge base to an INPUT file.

The INFER inference engine will read rules from this INPUT file.

Another example of a rule in the VAR knowledge base is as follows:

THIS IS RULE 6

CONFIDENCE LEVEL = 1.0

IF

The use of approximation concepts to reduce the dimensionality of the optimization problem is advised

AND

This problem is a truss problem

AND

The geometric configuration of this problem is fixed

AND

This problem is a statically determinate problem

THEN

This problem can have the inverse of cross-sectional area (1/A) as design variables. This would linearize stress and displacement constraints.

A major drawback of the current knowledge representation scheme that became evident in the testing stage is the frequency with which misrepresentation of knowledge can occur, particularly if text alone is used as a means of communication. Other forms of representation, in particular, representation by a graphical display, can represent some of the engineering knowledge more concisely.

In order to overcome this limitation, a graphical display function to represent the knowledge was developed within the INFER environment. An example of a rule in NODE knowledge base is the following:

```
THIS IS RULE 2

CONFIDENCE LEVEL = 1.0

IF

    &&& This problem has distributed loads

THEN

    &&& Choose nodes at all points where distributed
    loads change, and subdivision lines through these
    points are normal to boundary.
```

The "&&&" symbol is also a function symbol in the INFER inference engine. It can display the graphical form of knowledge for this condition or action on the computer terminal. The actual graphical display of this rule on the computer terminal is illustrated in Figure 4-8.

Another example of a condition in MESH2 knowledge base is as following:

```
&&& This problem has a quadrilateral domain or can be
considered a combination of several quadrilateral
domains: note that curved boundaries can be represented
by a combination of linear segments, and the analysis
domain may indeed be quadrilateral.
```

The actual graphical display of this condition on the computer terminal is shown in Figure 4-9.

Knowledge-Acquisition Facility

Knowledge-Acquisition Process

Since the current development was primarily intended as a concept demonstration system and one that would serve as a test bed for seeking improvements in expert system

THIS IS RULE 2
CONFIDENCE LEVEL= 1.0

IF

&&& This problem has distributed loads.

TO CONTINUE.....
PLEASE PRESS THE "ERASE" KEY TO CLEAN
THE SCREEN AND PRESS "RETURN" KEY

THEN

&&& Choose nodes at all points where distributed change, and subdivision lines through these points normal to boundary.

TO CONTINUE.....
PLEASE PRESS THE "ERASE" KEY TO CLEAN
THE SCREEN AND PRESS "RETURN" KEY

Figure 4-8. The actual graphical display of a rule on the computer terminal.

```
??
5

PLEASE INPUT THE FIRST AND LAST RULE NUMBER
THAT YOU WISH TO DISPLAY?
1 1

THIS IS RULE   1
CONFIDENCE LEVEL=  1.0

IF

&&& This problem has a quadrilateral domain or can
be considered a combination of several quadrilateral
domains: note that curved boundaries can be represented
by a combination of linear segments, and the analysis
domain may indeed be quadrilateral


TO CONTINUE.....
PLEASE PRESS THE "ERASE" KEY TO CLEAN
THE SCREEN AND PRESS "RETURN" KEY
```

Figure 4-9. The actual graphical display of a condition on the computer terminal.

methodology, the knowledge-acquisition process was a little different from what would be conventionally adopted. Prior experience in optimum structural synthesis was useful in consolidating information pertinent to the problem. This information was presented and discussed with a domain expert to validate its inclusion.

The other principal source of knowledge was the documentation available in the literature, such as books, journals, review articles, reports on specialized programs, and test problems with subject programs.

The knowledge-acquisition process is recognized as the most difficult task during this expert system development. After the initial efforts in eliciting knowledge for the OPSYN expert system, the idea of using existing CAD capabilities as a tool to elicit knowledge directly from the domain expert was proposed. Since most domain experts are likely to perform their design task in CAD-based systems, that system can play a crucial role in knowledge-acquisition process.

Knowledge-Acquisition System

An automated knowledge-acquisition system, embedded in a CAD environment, is developed in this study. The approach used in this system requires several domain experts to perform optimum design tasks on a set of prestructured problems. The domain expert performs those tasks with the program interactively. A combination of the confidence

levels associated with each response from the domain expert is stored in a file. The knowledge stored in this file can be analyzed to propose rules for the knowledge base. This approach is similar to the protocol analysis technique.

This knowledge-acquisition system includes two major components: the ACQU2 program to elicit knowledge from the domain experts and another program, RULE, to evaluate the knowledge to obtain rules, as shown in Figure 4-10.

The first program ACQU2 is embedded in a CAD system and contains a set of prestructured problems. In the current study, only finite element modeling task was considered for this purpose. An edge-loaded square plate problem with a hole in the domain was selected as an example problem. The shape of the hole can be a circle, an ellipse, or a rectangule. The position and dimension of this hole can be changed and defined by the designer in order to reflect every situation during the finite element modeling process, as shown in Figure 4-11. Another example is a one-bay, two-storey frame structure with concentrated and distributed static loads, as shown in Figure 4-12.

ACQU2 can be used as a CAD-based finite element modeling system or as a knowledge-acquisition system. The domain expert has the option of performing the modeling task in either of these modes. If the domain expert elects to perform the CAD finite element modeling task, the system performs like any common CAD system. It allows the domain expert to perform each of the necessary tasks sequentially.

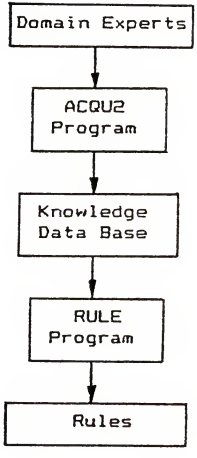


Figure 4-10. Architecture of a knowledge-acquisition system.

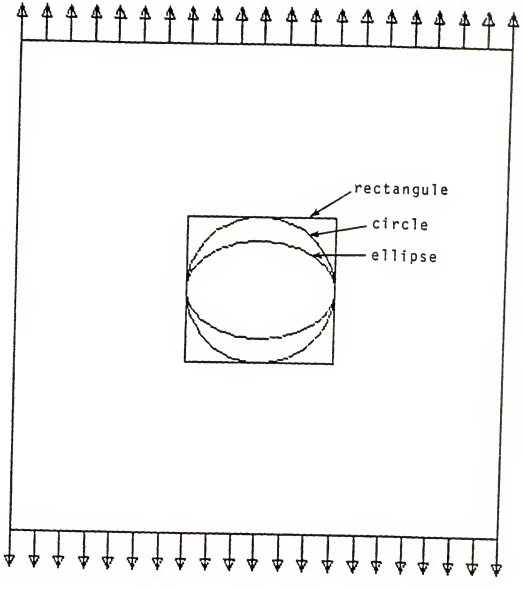


Figure 4-11. An edge-loaded square plate problem with different kinds of holes in the domain.

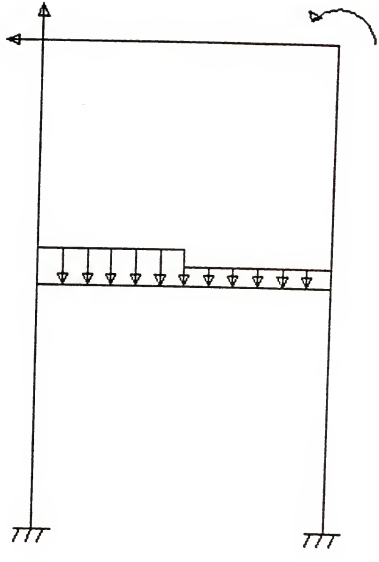


Figure 4-12. A one-bay two-storey frame structure with concentrated and distributed static loads.

These available tasks are: (a) selection of elements, (b) specification of nodes, (c) mesh generation, and (d) mesh refinement. In addition, for the purpose of flexibility, there are two other options, namely, "need more information" and "other choice," available for the domain expert to express his opinion on issues which are not addressed in the program.

The nodes in the model are generated by identifying the nodal coordinates at the very outset. After a few nodes are generated, the others may be entered by specifying the key node numbers and the relationship between the key nodes and the others, as shown in Figure 4-13. The mesh can be generated by identifying each element's node numbers or by specifying subdivision lines' node numbers, as shown in Figure 4-14.

The other mode in which the CAD-based ACQU2 program performs is as a knowledge-acquisition system for finite element modeling. In this mode, after the domain expert makes a task-related decision, the ACQU2 program queries the domain expert on the reasons for the decision and what confidence level can be associated with this decision. A set of responses to each task is predefined and the expert can choose one of these or provide a different response. The domain expert's response is then recorded in an IF-THEN rule form in a response storage file. Following are examples of the decisions made by the domain expert that are recorded in the storage file.

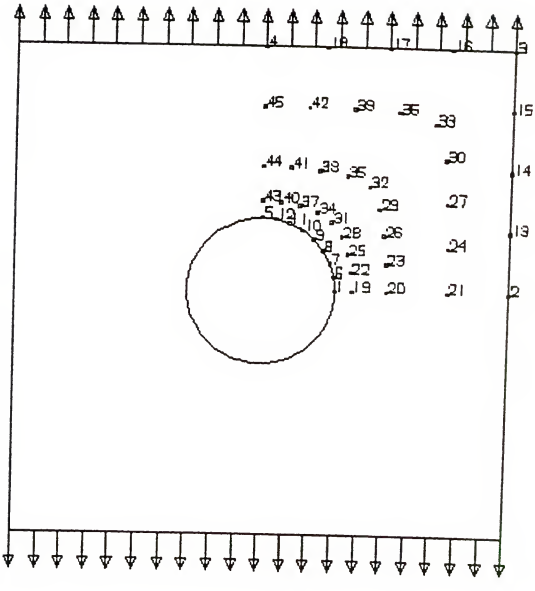


Figure 4-13. Nodes generation in finite element modeling.

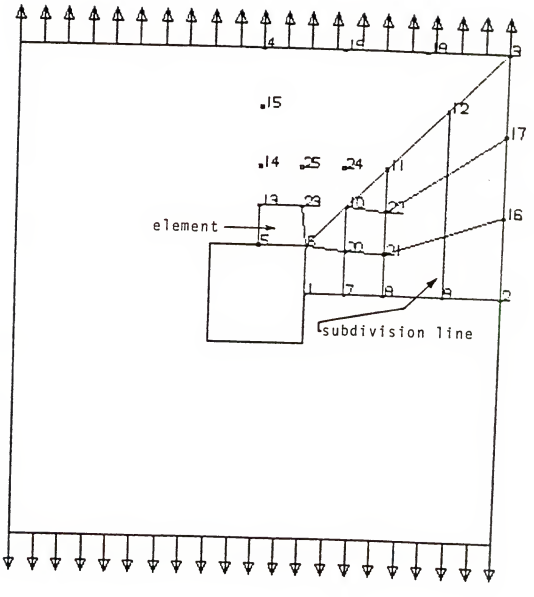


Figure 4-14. Mesh generation in finite element modeling.

```
CONFIDENCE LEVEL = 1.0

IF

    PLANE STRESS PROBLEM

THEN

    CHOOSE E41 QUADRILATERAL MEMBRANE ELEMENTS.

CONFIDENCE LEVEL = 0.9

IF

    BOUNDARY POSITION
AND
    THE BOUNDARIES CHANGE SHARPLY

THEN

    CHOOSE NODE AT 1 ( 250.0,  250.0)
    CHOOSE NODE AT 2 (1000.0, 1000.0).
```

The flowchart of the ACQU2 program is shown in Figure 4-15. A typical finished finite element model is shown in Figure 4-16.

The knowledge engineer then executes the second program RULE to analyze the knowledge in the file. The records in this file are transferred into the RULE program's data base by identifying each domain expert's name and each specified hypothesis. The records stored in the data base can be viewed as a three-dimensional decision table, as shown in Figure 4-17.

The RULE program evaluates these records and prints out lists of pertinent information for each rule. This information includes the number of domain experts that supported a rule, the average value of confidence level that

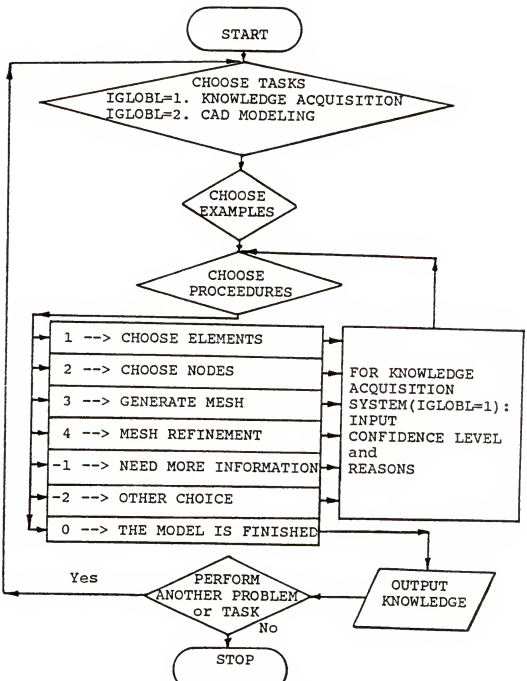


Figure 4-15. The flowchart of the ACQU2 program.

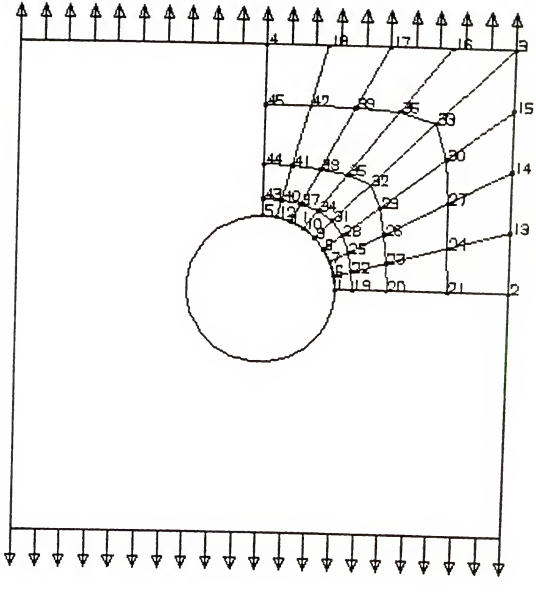


Figure 4-16. A typical finished finite element model.

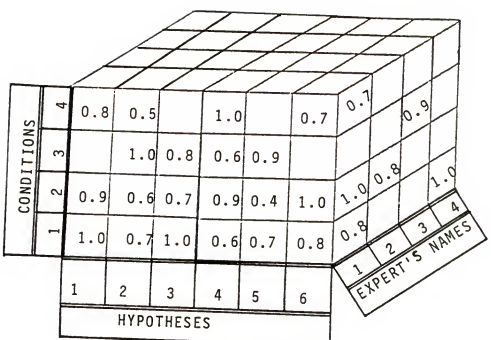


Figure 4-17. A three-dimensional decision table of domain experts' knowledge.

the experts assigned this rule, and the distribution of confidence levels associated with this rule. For example,

TOTAL VALUES OF CONFIDENCE LEVEL = 8.5
NUMBER OF EXPERT PERSONS = 10
AVERAGE CONFIDENCE LEVEL = 0.850

The distribution of confidence levels for a rule is shown in Figure 4-18. On the basis of this information for each rule, the knowledge engineer can select the best rule for a specified hypothesis.

In addition to transferring new records and evaluating records, there is another option available in the RULE program for the domain expert which is to modify the old records. The old records include those of the domain expert which are already part of the knowledge base. New records may be added as the domain expert gains new experience or different confidence levels may be assigned to those rules which were identified by other domain experts. This option provides a flexibility to update the rule data base and more complete information for the knowledge engineer to evaluate the rules.

The RULE program also has an option to give the knowledge engineer permission to scan each domain expert's records. If the range of variation of a domain expert's records of confidence level is less than 2/3 of total available range, then the program will allow the knowledge engineer the option of examining this domain expert's

Range of confidence levels for a given rule	Number of experts supporting the rule with this level of confidence
1.0 -> 0.9	6
0.9 -> 0.8	2
0.8 -> 0.7	1
0.7 -> 0.6	0
0.6 -> 0.5	1
0.5 -> 0.4	0
0.4 -> 0.3	0
0.3 -> 0.2	0
0.2 -> 0.0	0

Figure 4-18. The distribution of confidence levels for a rule.

program calculates the difference between the domain expert's record and the average of other domain experts' records for each rule. The values of differences are displayed to allow the knowledge engineer to determine if a skew exists in that particular expert's response and if a suitable weight factor should be added to the records. This option can prevent improper responses from the domain experts which might bias or distort the knowledge base. The flowchart of the second program is shown in Figure 4-19.

The preliminary testing of this knowledge-acquisition system verifies the usefulness of this approach. The availability of such a knowledge-acquisition system can help the knowledge engineer in eliciting knowledge from the domain experts and implementing the knowledge in the knowledge base.

The INFER inference engine and its environment for the OPSYN expert system development and the extent of knowledge base in the OPSYN expert system will be presented in Chapters 5 and 6, respectively.

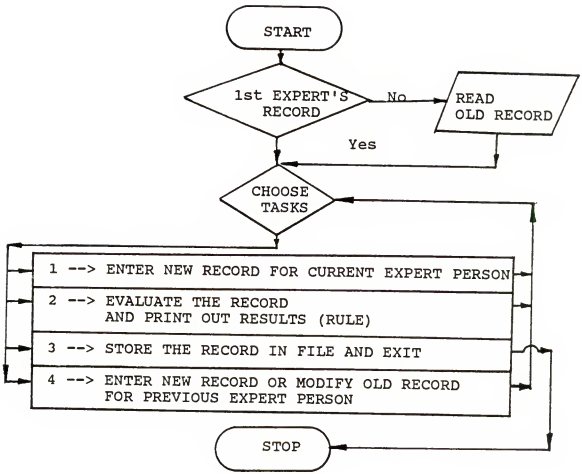


Figure 4-19. Flowchart of the RULE program.

CHAPTER V

THE INFER INFERENCE ENGINE

Introduction

Recently, many inference engines have been developed and are available in the form of commercial expert system shells for the purpose of developing expert systems. However, most available inference engines have not been developed for use in engineering design and analysis. Futhermore, the computer languages used in developing these inference engines are the languages especially developed for AI applications, such as LISP and PROLOG. Members of the engineering community who must build the systems with these shells feel more comfortable with languages such as PASCAL and FORTRAN.

Most current expert systems for engineering design and analysis have been developed in the environment of available commercial expert system shells. For example, SACON uses the EMYCIN [76] framework for its development. Experience with such systems has indicated that their utilization is severely limited unless they can be integrated with algorithmic analysis programs. The design or analysis of a structure requires more than heuristic insight and understanding of the problem. Mathematical models that

predict the behavior of structures under applied loads are important in obtaining detailed quantitative information that can considerably aid the heuristic process. In general, these mathematical models can be more efficiently implemented in algorithmic programs.

The requirements for an improved inference engine to be operational in engineering design and analysis problems are that it has the capability to execute algorithmic programs, an interface to extract results from the algorithmic programs output, and an interface to set up programs or input data for algorithmic programs.

At the inception of this study, an expert-system-development tool AESOP [31] was chosen as the frame for initial development and study. As the study progressed, the limitations of this tool became increasingly evident. The decision to build the INFER inference engine was based primarily on the need for eliminating the drawbacks of the AESOP inference engine.

The principal characteristics of the INFER inference engine and its environment can be described as follows:

(a) It uses a rule-based knowledge representation scheme with graphical display capabilities.

(b) It includes both forward-chaining and backward-chaining reasoning techniques, and an extensive explanation facility.

(c) It has a knowledge-acquisition system and a knowledge-base editor to assist the knowledge engineer in

the knowledge-elicitation process.

(d) It has the capability to deal with conventional programs.

(e) It can handle incomplete knowledge by associating confidence levels with the facts.

In this chapter, The INFER inference engine and its environment for expert system development are introduced. The architecture of the INFER inference engine and its environment is presented in the next section. Details of each component are introduced in later sections.

The Structure of INFER Environment

INFER is a domain-independent rule-based inference engine. The current structure of the INFER environment is shown in Figure 5-1. The INFER environment contains an inference engine with an explanation facility, a temporary facts data base, a knowledge-base editor, a knowledge-acquisition facility, and a user-friendly interface. It is written in FORTRAN-77, which is familiar to most structural engineers and makes communication between the expert system and the external algorithmic programs relatively straightforward. Currently, INFER is operational on a VAX 11-750 system.

INFER is especially developed for the building of expert systems for engineering design and analysis and is intended to be operational in a CAD environment. It has the capability to deal with algorithmic programs, such as finite

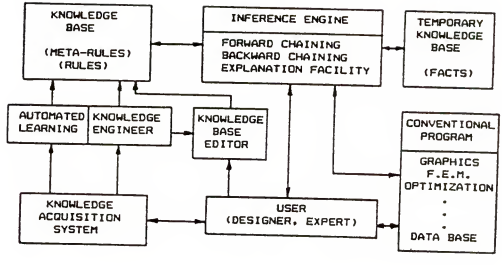


Figure 5-1. Structure of the INFER environment.

element analysis, optimization, graphics, data base manipulators, etc. This capability will be presented in greater detail in a later section.

Details of the knowledge-acquisition facility have been introduced in the previous chapter. Other components in INFER environment are presented next.

Inference-Reasoning Facility

Inference-Reasoning Techniques

The flowchart of the INFER inference engine is shown in Figure 5-2. During the inference-reasoning process, the first step is reading the rules in text and graphical format from the knowledge base. The step 2, if necessary, is to read the existing facts from the temporary facts data base. Choosing a suitable inference-reasoning technique is the third step, and INFER allows the user to make an appropriate selection. The available techniques are forward-chaining reasoning, backward-chaining reasoning, and special reasoning. The fourth step shown in the flowchart is the inference reasoning process itself. If a conclusion is confirmed and it contains a function symbol in the statement, then the program executes this function. After the reasoning process, the user has the option of seeking an explanation, performing another task, or exiting from the program. Details of the algorithm for each inference-reasoning technique are presented next.

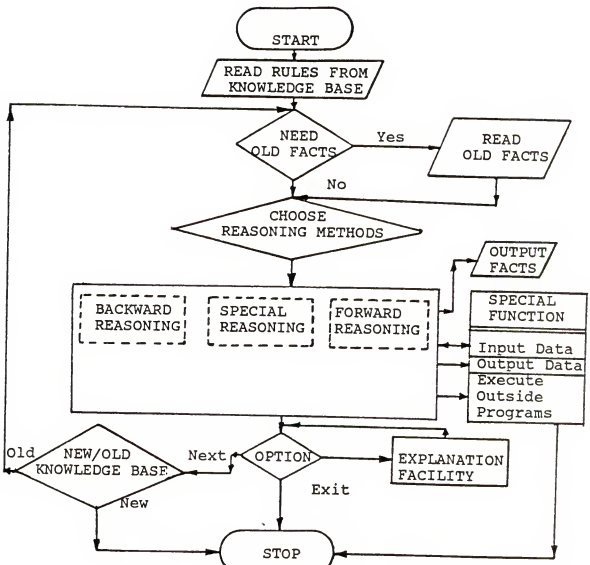


Figure 5-2. The flowchart of the INFER inference engine program.

For forward-chaining reasoning, the algorithm implementation in the INFER inference engine is summarized as follows:

1. Start from first rule's first condition.
2. Check this condition with existing facts. If a fact can be matched to this condition, then assign the confidence level of the fact to this condition and go to Step 3. Otherwise, assign a confidence level of 0.0 to this condition or ask the user to input a confidence level for this condition, and go to Step 3.
3. If all conditions of this rule have been checked, then compute the "AND" case confidence level for this rule,

   (confidence-level-of-this-rule)
   = (confidence-level-of-the-first-condition)
     * (confidence-level-of-the-second-condition)
     . . .
     * (confidence-level-of-the-last-condition)

   and go to Step 4. Otherwise, switch to next condition in this rule and return to Step 2.
4. If the confidence level of this rule is 0.9 or higher, then display all actions of this rule to the user and record them in the facts data base and return to the first step. Otherwise, check the other rules' action parts with this rule's action parts and go to Step 5.

5. If no other rule's action parts matches with this rule's action parts, then go to Step 7. Otherwise, go to Step 6.

6. If all rules that have the same action parts are checked, then compute the "OR" case confidence level for these rules,

```
new-confidence-level = confidence-level-one
  + ((1.0 - confidence-level-one)
     * confidence-level-two)
```

and assign the confidence level to the last rule and return to Step 4. Otherwise, switch to the first condition of one of above rules that have not been checked and return to Step 2.

7. If all rules have been checked, terminate the reasoning. Otherwise, switch to the first condition of the next rule that has not been checked and return to Step 2.

A step by step description of the algorithm of backward-chaining reasoning in the INFER inference engine is as follows:

1. Start from first rule's hypothesis.

2. Check this hypothesis with existing facts. If a fact matches this hypothesis, then assign the confidence level of this fact to this hypothesis and go to Step 6. Otherwise, start from the first condition of this hypothesis and go to Step 3.

3. Check this condition with existing facts. If a fact matches with this condition, then assign the confidence level of this fact to this condition and go to Step 5. Otherwise, check this condition with other hypotheses within the knowledge base and go to Step 4.
4. If a hypothesis matches this condition, then assign this hypothesis as a sub-hypothesis and switch to the first condition of this sub-hypothesis and return to Step 3. Otherwise, ask the user to input the confidence level for this condition and go to Step 5.
5. If all conditions of this hypothesis (or sub-hypothesis) have been checked, then compute the "AND" case confidence level for this hypothesis (or sub-hypothesis) and go to Step 6. Otherwise, switch to the next condition in this hypothesis (or sub-hypothesis) and return to Step 3.
6. If the confidence level of this hypothesis (or sub-hypothesis) is 0.9 or higher, then display the action parts to the user and record the action parts into facts data base and go to Step 7. Otherwise, check this hypothesis (or sub-hypothesis) with other rules' hypotheses and go to Step 8.
7. If this is a hypothesis, terminate. Otherwise, assign the confidence level of this sub-hypothesis

to the corresponding condition in the upper level hypothesis, switch to this condition and return to Step 5.

8. If the hypotheses of no other rule matches this hypothesis (or sub-hypothesis), then go to Step 10. Otherwise, go to Step 9.
9. If all other rules that have the same hypothesis as this hypothesis (or sub-hypothesis) have been checked, then compute the "OR" case confidence level for these hypotheses (or sub-hypotheses) and assign this confidence level to the last hypothesis (or sub-hypothesis) and go to Step 6. Otherwise, switch to the first condition of one of above rules that had not been checked and return to Step 3.
10. If all hypotheses have been checked, terminate. Otherwise, switch to the next hypothesis that had not been checked and return to Step 2.

Special reasoning is similar to backward-chaining reasoning except that all rules are examined even if conclusions were already confirmed by using only part of the rules. This means that the "terminate" action in Step 7 of the backward-chaining reasoning algorithm is replaced by "go to Step 10" action for the special reasoning algorithm. The availability of the special reasoning algorithm is necessary for tasks like nodes and subdivision lines selection in finite element modeling or constraints selection in optimum design modeling. These tasks are of the multi-choice type

that require the designer to consider every available choice. For example, the task of constraint selection for a structural optimization problem can result in any one of various combinations, such as stress, stress and displacement, stress and natural frequency, etc.

Explanation Facility

The INFER allows the designer to respond with confidence levels, and also provides an explanation facility. In the event that a condition exceeds a confidence level of 0.9 (or 90%), it is confirmed as the optimum conclusion. If all the rules are examined and no confirmed condition is obtained, the status of all conditions with a confidence level of 0.1 (or 10%) or higher can be displayed in an explanation facility. An example of the display of explanations is as follows:

```
THIS IS RULE 6
YOU HAVE 1.0 CERTAINTY FOR -->
   The use of approximation concepts to reduce the
   dimensionality of the optimization problem is advised
YOU HAVE 1.0 CERTAINTY FOR -->
   This problem is a truss problem
YOU HAVE 0.9 CERTAINTY FOR -->
   The geometric configuration of this problem is fixed
YOU HAVE 0.9 CERTAINTY FOR -->
   This problem is a statically determinate problem
SO THE FOLLOWING HYPOTHESIS HAS 0.81 CERTAINTY -->
```

This problem can have the inverse of cross-sectional area (1/A) as design variables. This would linearize stress and displacement constraints.

This display permits the user to choose those conditions with the highest level of confidence and to understand how the INFER inference engine arrived at those conclusions.

Knowledge-Representation Scheme

The knowledge-representation scheme in the INFER environment is an IF-THEN rule-based scheme. Details of examples of this knowledge-representation scheme have been presented in the previous chapter where it was noted that the knowledge base was designed to be separated into different modules for using suitable inference-reasoning techniques and to speed up the inference process. The rules in each knowledge base are assigned a rule order number. The inference engine executes the rules according to the order of rule number by starting from the lowest rule number. The rules in the knowledge base are stored according to the order of editing rather than the order of rule number.

The INFER also has a graphical display function to represent the knowledge. Such graphical displays can represent some engineering knowledge more concisely than conventional text-based representation. This function is embedded in a CAD environment and is introduced in the last chapter. The graphical data are stored together with the rules in the knowledge base and is identified by using the

associated rule number as well as condition or action number.

A temporary facts data base facility is contained in the INFER environment. This facility is configured to collect the facts or deductions as they become available during the execution of the INFER inference engine, into a file "FACTS". By using the information in this FACTS file in the inference-reasoning process, established facts become common to other modules of the knowledge base. For the knowledge base containing meta-rules, the condition parts of this knowledge base will not be recorded into the FACTS file, in order to avoid the infinite loop in using meta-rules for selection of other rules.

Knowledge-Base Editor

The INFER environment has a knowledge-base editor facility to help the knowledge engineer in editing and modifying the rules in the knowledge base. This facility would shorten the expert system development time and the learning time for a new user.

This knowledge-base editor program KBSED can perform a variety of tasks, such as creating new rules, modifying old rules, deleting old rules, changing rule order, displaying rules, changing knowledge base type, and updating the knowledge base. The task of modifying old rules includes changing confidence levels, adding conditions or actions, deleting conditions or actions, and modifying conditions or

actions. The knowledge base types can be divided into meta-rules knowledge base, the knowledge base containing rules in a graphics format, and common rules knowledge base. The task of updating knowledge base deletes the INPUTKB file and creates the NEW knowledge base file, which can be assigned to the specified file name in the knowledge base. KBSED also has the capability to create and modify graphical data. The flowchart of this program is shown in Figure 5-3.

Special Functions

Since INFER was developed for engineering analysis and design, it has the capability to communicate with conventional engineering algorithmic programs. Data can be transferred both to and from INFER to such programs. Further, INFER can generate commands to execute external programs. Those special functions are shown schematically in Figure 5-4.

The INFER inference engine can recognize the "***" symbol at the beginning of each action statement and generate a command file "OUT" which contains the statement. The user then has the option of executing this command file. This function is used in the meta-rules to assign the file in the knowledge base to an "INPUT" data file. It also can be used to execute external programs. An example of the statement in an action part of a rule to run the external program OPTIMAL is as follows:

"***RUN OPTIMAL"

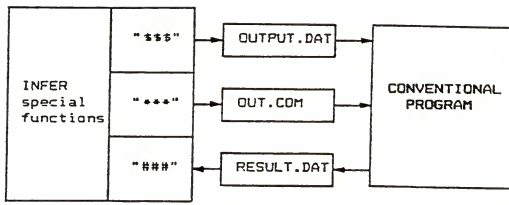


Figure 5-3. The flowchart of the KBSED program.

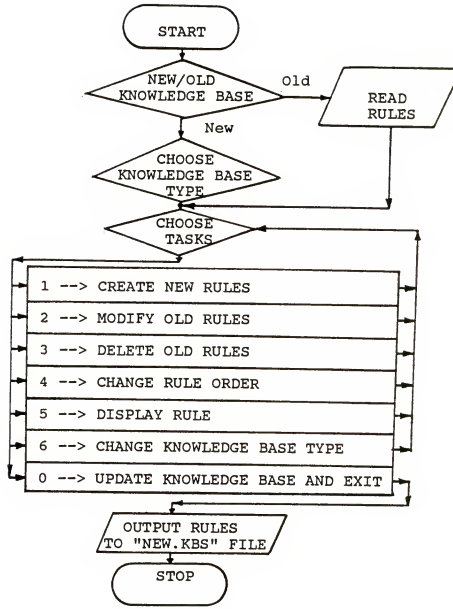


Figure 5-4. Available special functions in the INFER inference engine.

The "$$$" symbol at the beginning of an action statement is the function symbol for INFER to generate the output data file "OUTPUT", which contains the statement in this action. The expert systems can use this function to send information or data to external programs. An example of using the "$$$" symbol to send a FORTRAN statement to the OUTPUT file is as following:

```
"$$$100    IF(A.EQ.B)C=0.0"
```

The function symbol for the INFER inference engine to extract information or data from external programs is the "###" symbol at the beginning of a condition or action. It requires the external program to assign the information or data to the special data file "RESULT". The INFER inference engine reads the information or data from the RESULT file and stores them in the FACTS file. The information or data in the RESULT file are arranged according to the rule form in the INFER inference engine.

With these functions, expert systems and conventional programs can be combined together to obtain intelligent CAD systems, as shown schematically in Figure 4-1. The availability of these functions also enhances the role of the INFER inference engine to serve as an environment in the development of expert systems for engineering design and analysis.

Illustrative examples will be presented in Chapter 7 to demonstrate the use of INFER in the OPSYN expert system.

Details of the knowledge in each module of the knowledge base of the OPSYN expert system will be introduced in the next chapter.

CHAPTER VI

EXTENT OF KNOWLEDGE BASE

Introduction

In this chapter, details of the information contained in the OPSYN expert system's knowledge base are presented. As mentioned in Chapter 4, the knowledge base is separated into three modules, namely, FEMOD, OPDEM, and OPPER for the tasks of finite element modeling, optimum design modeling, and optimization performance, respectively. A list of rules in the OPSYN expert system can be obtained in Reference 77. In the present form, there are about 264 rules in the knowledge base. The numbers of rules contained in the FEMOD, OPDEM, and OPPER modules are 87, 118, and 59, respectively. The scope of each module of the knowledge base is presented in the following sections.

Finite Element Modeling

The knowledge base consists of both rules for finite element modeling and a particular set of rules for the capabilities of a finite element program EAL. The tasks of finite element modeling include element-type selection, node and subdivision-lines selection, mesh generation, and mesh

refinement. The scope of knowledge base for each of these tasks is presented next.

The rules for element selection in this knowledge base are applicable to the finite element program EAL, and refer to the elements available in the program library. A detailed description of the element available in the EAL program is given in Appendix A. The FEMOD module assists in the selection of elements for a problem based on the following considerations: (a) whether the type of problem is a mechanism or a true load-carrying structure; (b) whether the structure can be visualized as a truss or needs to be analyzed as a beam frame assembly; (c) whether the structure can be construed to be in plane stress or plane strain, and which in turn determines if it can be modeled by membrane, plate, or shell elements; (d) whether the failure mode of the problem is yielding or buckling; (e) whether the problem pertains to static loading only or if it requires dynamic load analysis and frequency computations; (f) whether the problem is isotropic or laminate plate; and (g) special considerations for problems that have specific deflection, or are subjected to shear loading, torsional loading, axial loading, or inplane/out-of-plane loading.

In order to determine the location of node points and subdivision lines in the finite element model one must consider the following aspects: (a) the location of concentrated loads in the structure, and, for distributed loads, the position where the load magnitude changes

appreciably; (b) the existence of material inhomogeneity in the analysis domain; (c) the presence of irregular domain boundaries; (d) the presence of a geometric and loading symmetry in the structure; (e) select locations in the analysis domain where analysis results are required; (f) change in cross sectional properties; (g) change of boundary conditions; and (h) the existence of holes or discontinuities in the analysis domain.

In the context of mesh generation, the decision pertaining to the size and distribution of elements must consider the following points: (a) the existence of complicated domain boundaries; (b) the type of failure mode appropriate to the structure; (c) the type of loading that the structure is subjected to--concentrated or distributed loads; (d) the existence of holes or discontinuities in the analysis domain; (e) an analysis that is primarily focussed on displacement or if both displacement and stresses are required; and (f) the selection of element shapes (i.e., triangular or quadrilateral) to account for irregular domains and boundaries.

For mesh refinement problems, special consideration was given to the following: (a) whether the mesh arrangement is appropriate to minimize the element distortion; (b) whether the element aspect ratio is within a permissible range to achieve satisfactory results; (c) whether a change in the node numbering sequence would reduce bandwidths, and hence also reduce the computational effort; and (d) whether the

analysis results display a smooth spatial variation over the domain. The presence of stress concentration areas would mandate a rezoning of the problem domain to account for the concentration effects.

Optimum Design Modeling

Although the knowledge base is separated into ten files, the rules for optimum design modeling can be divided into the following categories: (a) meta-rules; (b) rules for formulating an efficient design model; (c) rules for formulating the optimization problem--selection of design variables, constraints, and objective function; and (d) rules for methods selection--selection of approximation techniques, strategies, and sensitivity analysis methods. The scope of knowledge base for each of these categories is presented next.

Meta-Rules

The meta-rules lead the designer to rules for efficient model formulation at the beginning of design modeling. If the problem requires multilevel decomposition techniques, then each subproblem must reconsider the rules for efficient model formulation again. At the next step, the designer has the freedom to decide the sequence of choosing design variables, constraints, and objective function in these meta-rules. Finally, the designer executes the rules for

choosing approximation methods, optimization strategies, and design sensitivity methods.

Rules for Formulating an Efficient Design Model

Here, the tradeoff between the desired solution accuracy and reduced computational times are the two key points under consideration. The possibility of generating a simplified model for optimum design, or dimensionality reduction of the detailed design space by design variable linking, efficient constraint representation, and approximation concepts are considered in the knowledge base. Another possible alternative is the use of multilevel decomposition methods.

Additionally, the techniques of decomposing the problem are based on the following considerations: (a) whether the problem contains different disciplines; (b) whether the structure contains several substructures or can be divided into many substructures; and (c) special classes of problems such as composite sandwich-panels or beam/frame problems.

Rules for Design Model Formulation

For design variable selection, the following aspects are considered: (a) the types of problems are truss, beam, frame, membrane, shear panel, plate bending, composite plate, or shell structure; (b) the geometry of structure is fixed or subject to change; and (c) the use of approximation concepts. For optimum design problems using approximation

concepts, the selection of design variables needs the following additional information: (a) whether the truss, beam, or frame structure is statically determinate, mildly indeterminate, or strongly indeterminate; (b) whether the shell problem is under in-plane or out-of-plane loading; and (c) for a problem that deals with vibration analysis or is subjected to dynamic loading, the order of magnitude of nonstructural mass to structural mass.

In identifying constraints, one must consider the following points: (a) the type of failure mode appropriate to the structure; (b) the type of loading that the structure is subjected to--static vs. dynamic; (c) geometrical restrictions on design variables; (d) consider constraints on deflection or the stiffness; (e) special constraints that may be encountered in aeroelastic synthesis; (f) weight is not the principal objective function and must be considered as a constraint; and (g) the use of approximation concepts in buckling analysis.

For objective function selection, the rules in the knowledge base seek to determine if, (a) the design problem has one important criterion or if multiple criteria must be considered; (b) whether the most important criterion is weight; and (c) whether the problem uses multilevel decomposition methods.

Futhermore, for checking the formulation of a design model, one must consider the following points: (a) the objective function and each constraint must depend on some

or all of the design variables; (b) design variables must be independent of each other; and (c) design variables must be suitably selected in an attempt to linearize the constraints first and then the objective function.

Rules for Methods Selection

The rules for approximation technique selection must address the following concerns: (a) whether some of the independent design variables are similar to others or have fixed relationship to others; (b) whether the optimization problem has a large number of constraints; (c) whether the finite element model has a large number of degrees of freedom; and (d) special consideration for the allowable move limit on design variables.

For the selection of appropriate strategies, one must consider the following aspects: (a) the objective function and constraints can be represented in a generalized positive polynomial form; (b) the problem is separable or can be approximated well as a separable problem; (c) the problem has a quadratic but nonseparable objective function and linear inequality constraints; (d) the problem has discrete design variables; (e) the problem has only stress constraints; and (f) the use of approximation concepts.

In selecting sensitivity analysis methods, special attention must be given to the following: (a) whether the optimization problem is a member resizing or shape optimization problem; (b) whether the computational time is

an important issue; (c) whether the number of active constraints is expected to be less than the number of design variables multiplied by the number of loading conditions; and (d) special consideration for the type of constraints--static stress, static displacement, vibration, dynamic response, or flutter constraints.

Optimization Performance

Currently, the knowledge in the OPPER knowledge base module has only the capability of assisting the designer in selecting the best algorithm. The available general-purpose optimization program adapted for the current task is ADS version 1.10 [75]. This version contains the addition of a new convex linearization strategy and equality constraints to all options in the program. The description of the available methods in the ADS program are outlined in Appendix B.

Unlike the earlier effort [30], the OPPER module emphasizes the selection of best algorithms for the structural optimization problem. Since most structural optimization problems are constrained problems, the present approach considers the applicable methods for constrained problems and for constrained problems that are treated as unconstrained problems. Furthermore, the OPPER module also contains some knowledge for assisting the designer to switch to another algorithm when the previously-selected algorithm fails to converge to an acceptable solution.

According to the arrangement of the ADS program, the knowledge in the OPPER module can be divided into three categories: meta-rules and rules for methods selection, rules for constrained problems treated as unconstrained problems, and rules for constrained problems. The scope of the knowledge in each of these categories is described next.

Meta-Rules and Rules for Methods Selection

The meta-rules lead the designer to rules for methods selection at the beginning of optimization algorithms selection. After selecting the method, a designer then executes the rules for choosing a strategy, an optimizer, and a one-dimensional search method. If there is difficulty in obtaining a solution with the selected mix of approaches, then the designer can execute the rules for adjusting algorithm parameters or switching to other methods.

The tradeoff between efficiency and reliability is the principal goal for selection of the methods for constrained problems or the methods for constrained problems treated as unconstrained problems. For methods selection, the following aspects are also considered: (a) whether analytical gradients are available for this problem or not; (b) whether this problem has the possibility of local minima; (c) whether the objective function and/or constraints are nonlinear or strongly nonlinear; and (d) whether the evaluation of objective function and constraints are expensive (i.e., implicit functions).

The rules for adjusting algorithm parameters and for switching to other algorithms in the knowledge base are strongly dependent on which strategy is in use and which of the applicable parameters is likely to affect the convergence behavior.

Rules for Constrained Problems Treated as Unconstrained Problems

For strategy selection, the rules in the knowledge base consider the following aspects: (a) whether intermediate designs may have violated constraints; (b) whether constraint values increase sharply in the feasible domain; and (c) whether the final design must be feasible, can have slightly violated constraints, or must have precisely satisfied constraints (equality constraints).

In selecting an optimizer, the decisions obtained from the expert system take into account the following: (a) whether the available computer storage is expected to be limiting for the problem; (b) whether analytical gradients are available or not; (c) whether the analysis is iterative in nature; (d) whether the problem affords the possibility of generating an ill-conditioned objective function and/or constraints; and (e) whether this problem involves several design variables.

The rules for one-dimensional search methods selection are based on the following considerations: (a) whether this problem has an ill-conditioned or a well-conditioned

objective function and/or constraints and (b) whether function evaluations are overly expensive.

Rules for Constrained Problems

In selecting strategy, the following aspects are considered: (a) the objective function and/or constraints are linear or at most, mildly nonlinear; (b) the objective function is quadratic or nearly quadratic; (c) there are fewer or as many active constraints as design variables expected at the optimum; (d) intermediate designs may violate some of the constraints or must satisfy all constraints; (e) whether function evaluations are expensive or not; (f) whether relative minima are likely to exist or not; (g) whether reducing computational time is important for this problem; and (h) whether approximation concepts are used to linearize the objective function and constraints.

In comparison to the case of constrained problems treated as unconstrained problems, there are a few differences in the rules for optimizer selection to be considered, such as the following: (a) whether the objective function and/or constraints are linear or at most, mildly nonlinear; (b) whether there are fewer or as many active constraints as design variables expected at the optimum; (c) whether the analysis is iterative; and (d) whether function evaluations are computationally expensive.

For selection of the one-dimensional search method, the rules are based on the following information about the

problem: (a) whether the objective function and/or constraints are nonlinear; (b) whether function evaluations are expensive; (c) whether the problem has an ill-conditioned or a well-conditioned objective function and/or constraints; (d) whether there are fewer or as many active constraints expected at the optimum as the number of design variables; (e) whether this problem has an objective function and/or constraints which are nearly linear or quadratic; (f) whether this problem is a convex problem; (g) whether relative minima are likely to exist; and (h) whether this problem requires a strategy selection 0 (directly select an optimizer).

Illustrative examples will be presented in the next chapter to demonstrate the use of the OPSYN expert system.

CHAPTER VII

ILLUSTRATIVE EXAMPLES USING OPSYN SYSTEM

Introduction

As an illustration, five simple examples are presented in this chapter to show how the OPSYN expert system is used. The first and second examples demonstrate the use of the FEMOD module for finite element modeling. The use of the OPDEM module for optimum design modeling is presented in the third and fourth examples. The last example shows the use of OPPER in providing assistance for enhancing optimization performance. Only the first example is presented by a detailed map of an interactive session. Other examples are given by abbreviated maps of the interactive session. The underlined words and phrases in each session are the user response.

Finite Element Modeling

The first example is a one-bay, two-story frame structure with concentrated and distributed static loads, as shown in Figure 4-12. The frame is designed for minimum weight and subject to maximum allowable stresses in the members. A detailed map of an interactive session is shown in Figure 7-1.

```
$ COPY FEMOD.KBS;1 INPUT.DAT;1
$ RUN INFER

  DID YOU ASSIGN THE KNOWLEDGE BASE FILE AS "INPUT" FILES?
  OR COPY TO "INPUT.DAT" FILE?
  1 -->YES, 0 -->NO.
  1

            ***READING RULES***

  DO YOU WISH TO USE FACTS IN "FACTS.DAT" FILE?
  1 -->YES, 0 -->NO.
  0

  PLEASE CHOOSE INFERENCE PROCESS?
  1 --> FORWARD CHAINING (YOU ALREADY KNOW SOME FACTS!)
  2 --> BACKWARD CHAINING
  3 --> SPECIAL REASONING (ASK QUESTIONS FOR EVERY RULE!)
  2

            ***BACKWARD CHAINING PROCESS***

  PLEASE ANSWER THE FOLLOWING QUESTION WITH A
  FACTOR RANGING FROM
  0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
  ??
  The mesh has not been generated for this problem.
  1.0

  PLEASE ANSWER THE FOLLOWING QUESTION WITH A
  FACTOR RANGING FROM
  0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
  ??
  The FEM elements have not been selected for this problem.
  1.0

  ************************************
  THE HYPOTHESIS THAT HAS CERTAINTY 1.000

  You should run the ELEMENT knowledge base to choose
  elements for your problem by using backward-chaining
  reasoning.
  ***COPY ELEMENT.KBS;1 INPUT.DAT;3

  (The "***" symbol is a function symbol.  It can create a
   command file which essentially contains the statement
   following the symbol in the rule, and execute this
   command file.)
```

Figure 7-1. An interactive session of using FEMOD module for a bay-frame structure problem.

```
 PLEASE INPUT 1 TO RUN THIS PROGRAM?
 1
$ RUN INFER

 DID YOU ASSIGN THE KNOWLEDGE BASE FILE AS "INPUT" FILE?
 OR COPY TO "INPUT.DAT" FILE?
 1 -->YES, 0 -->NO.
 1

          ***READING RULES***

 DID YOU WISH TO USE FACTS IN "FACTS.DAT" FILE?
 1 -->YES, 0 -->NO.
 1

 PLEASE CHOOSE INFERENCE PROCESS?
 1 --> FORWARD CHAINING (YOU ALREADY KNOW SOME FACTS!)
 2 --> BACKWARD CHAINING
 3 --> SPECIAL REASONING (ASK QUESTIONS FOR EVERY RULE!)
 2

          ***BACKWARD CHAINING PROCESS***

 PLEASE ANSWER THE FOLLOWING QUESTION WITH A
 FACTOR RANGING FROM
 0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
 ??
 This problem is a mechanism problem.
 0.0

 PLEASE ANSWER THE FOLLOWING QUESTION WITH A
 FACTOR RANGING FROM
 0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
 ??
 This problem is beam or frame.
 1.0

 PLEASE ANSWER THE FOLLOWING QUESTION WITH A
 FACTOR RANGING FROM
 0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
 ??
 Only one plane of bending or axial force for this problem.
 0.5

 PLEASE ANSWER THE FOLLOWING QUESTION WITH A
 FACTOR RANGING FROM
 0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
 ??
 This problem has applied torsional force.
 0.0
```

Figure 7-1--continued

```
PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
This problem is truss structure.
0.0

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
Only extensional vibration for this problem.
0.2

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
Only axial force for this problem.
0.0

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
The stiffness matrix for this problem is already known.
0.5

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
This problem has more than one plane of bending and any
combination of axial, bending and torsional loads.
0.9

**************************************
THE HYPOTHESIS THAT HAS CERTAINTY 0.900

This problem requires elements E21.  The E21 are general
straight or circularly curved beam elements, such as
channels, wide-flanges, angles, tubes, zees.

OPTIONS ARE
 1 --> SEE AN EXPLANATION
 2 --> CONTINUE TO NEXT CASE
 3 --> EXIT
? PLEASE ANSWER WITH THE NUMBER !
1
```

Figure 7-1--continued

```
THIS IS RULE 8

YOU HAVE 1.00 CERTAINTY FOR -->
This problem is beam or frame.

YOU HAVE 0.90 CERTAINTY FOR -->
This problem has more than one plane of bending and any
combination of axial, bending and torsional loads.

SO THE FOLLOWING HYPOTHESIS HAS 0.900 CERTAINTY -->
This problem requires elements E21.  The E21 are general
straight or circularly curved beam elements, such as
channels, wide-flanges, angles, tubes, zees.

PLEASE ENTER RETURN KEY TO CONTINUE !!

OPTIONS ARE
 1 --> SEE AN EXPLANATION
 2 --> CONTINUE TO NEXT CASE
 3 --> EXIT
? PLEASE ANSWER WITH THE NUMBER !
3

          ***THANK YOU !!***

THE SYSTEM IS DEALING WITH THE META-RULES KNOWLEDGE BASE.
PLEASE RUN INFER PROGRAM AGAIN TO CONTINUE.
FORTRAN STOP
$ RUN INFER
    .
    .
    .
PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
The FEM elements have been selected for this problem.
1.0

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
This structure is modeled by one-dimensional line
elements.
0.9
```

Figure 7-1--continued

```
**************************************
THE HYPOTHESIS THAT HAS CERTAINTY 0.900

You should run the MESH1 knowledge base to generate mesh
by using backward-chaining reasoning.
***COPY MESH1.KBS;1 INPUT.DAT;3

PLEASE INPUT 1 TO RUN THIS PROGRAM?
1
$ RUN INFER
   .
   .
   .
PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
This problem is under dynamic loading or requires
computation of lateral deflections.
0.3

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
This problem considers buckling case.
0.7

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
This problem is only under static loading.
1.0

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
This problem only considers yielding case.
0.95

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
This problem is only under concentrated loading.
0.1
```

Figure 7-1--continued

```
PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
This problem has distributed loading.
1.0

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
This problem has axes of symmetry for both the geometry
and loading.
0.0

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") to 1.0 (DEFINITE "YES")
??
This problem is unsymmetric in geometry or loading.
1.0

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
This problem is a beam problem.
0.5

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
This problem only needs displacement informations.
0.5

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
This problem needs stress informations.
1.0

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
This problem is frame problem.
1.0
```

Figure 7-1--continued

```
**************************************
THE HYPOTHESIS THAT HAS CERTAINTY 0.950

Choose nodes at following places: support positions, joint
locations, at each end of the frame member, at
concentrated loads position and points where the
distributed loads change significantly, also choose nodes
where response may be required.  If distributed loads
range is long, choose some nodes in the range of the
distributed loads.  Also choose 5 to 10 elements for each
frame member and try to make the element lengths equal.

OPTIONS ARE
 1 --> SEE AN EXPLANATION
 2 --> CONTINUE TO NEXT CASE
 3 --> EXIT
? PLEASE ANSWER WITH THE NUMBER !
3
   .
   .
   .
```

Figure 7-1--continued

The finite element model based on the recommendation from the FEMOD module is shown in Figure 7-2.

The results obtained from the OPSYN system vary as the response input confidence level is changed. For example, in choosing the elements for this problem, the input of confidence level for the condition "Only extensional vibration for this problem" is "0.2." Since this structure is subjected only to static loads, the change of the confidence level for this condition from "0.2" to "0.0" will not affect the result. As for another condition, "This problem is beam or frame," the change of confidence level from "1.0" to "0.8" will also influence the result. Instead of the conclusion shown in Figure 7-1, the result will be changed to "no conclusion for this problem." However, the designer still can receive some recommendation by using the explanation facility.

The second example is a classical test problem of an edge-loaded, thin plate with a circular hole, as shown in Figure 7-3. The application of the OPSYN expert system in the task of finite element modeling results in the following system response, as shown in Figure 7-4.

The finite element model for this problem is shown in Figure 7-5.

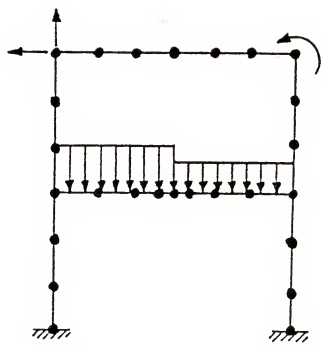


Figure 7-2. The finite element model for the bay-frame structure problem.

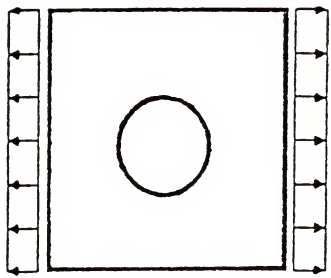


Figure 7-3. An edge-loaded, thin plate with a circular hole.

```
$ COPY FEMOD.KBS;1 INPUT.DAT;1
$ RUN INFER
     .
     .
     .
  ??
  This problem is a plane stress problem.
  0.8

  PLEASE ANSWER THE FOLLOWING QUESTION WITH A
  FACTOR RANGING FROM
  0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
  ??
  This problem is such that only inplane loads are applied,
  and that there are no rotational degrees of freedom
  permitted at any point of structure modeled by these
  elements.
  0.7

  ****************************************
  THE HYPOTHESIS THAT HAS CERTAINTY 0.940

  This problem requires elements E31 or E41.  The E31 are
  triangular membrane elements.  The E41 are quadrilateral
  membrane elements.
     .
     .
     .
$ RUN INFER
     .
     .
     .
  ??
  This problem has symmetrical geometry and loading.
  1.0

  ****************************************
  THE HYPOTHESIS THAT HAS CERTAINTY 1.000

  Symmetry indicates a reduced analysis problem: one or two
  axes of symmetry would require half or quarter domain.
  Additional lines of symmetry dictate a corresponding
  reduction in the analysis domain.  Choose subdivision
  lines along axes of symmetry and nodes on these lines.
     .
     .
     .
  ??
  The domain has holes or cutouts.
  0.9
```

Figure 7-4. An interactive session of using FEMOD module for a plate problem.

```
**************************************
THE HYPOTHESIS THAT HAS CERTAINTY 0.900

You need a refined mesh for the domain in the proximity of
the discontinuity.
For static analysis problems: choose nodes in the domain
around the hole, at least 2 to 5 layers around the hole,
the mesh-size-ratio (typically defined as the size of a
cell closest to discontinuity to the cell size in the next
layer of elements) should be about 0.5.  For the narrow
domain between the outer and inner boundary, you should
also consider a refined mesh.
    .
    .
    .
$ RUN INFER
    .
    .
    .
??
&&& This problem has a quadrilateral domain or can be
considered a combination of several quadrilateral domains:
note that curved boundaries can be represented by a
combination of linear segments, and the analysis domain
may indeed be quadrilateral.
0.9

(The "&&&" symbol is a function symbol.  It can display
 the graphical form of knowledge for this condition or
 action on the computer terminal.)
    .
    .
    .
??
This problem considers only yielding case.
1.0
    .
    .
    .
??
This problem is under only static loading.
1.0
```

Figure 7-4--continued

```
**************************************
THE HYPOTHESIS THAT HAS CERTAINTY 0.900

&&& Wherever possible use quadrilateral elements.  Use
triangular elements if domain boundary is such that these
elements are required or if these elements can be used in
conjunction with quadrilateral elements to maintain any
inherent symmetry in the problem.  Using either
quadrilateral or triangular elements, it is important to
retain any mathematical symmetry that may exist, that is
both loading or support symmetry.  Number of elements
should be such that there are at least 4 to 8 nodes along
the longer side and 3 to 4 nodes along the shorter
dimension.  For a square domain use at least 4 to 8 nodes
along each side.
   .
   .
   .
$ RUN INFER
   .
   .
   .
??
The mesh has been generated for this problem.
1.0

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
You want to check element distortion, aspect ratio, or
node numbering.
0.9

**************************************
THE HYPOTHESIS THAT HAS CERTAINTY 0.900

You should run the SHAPE knowledge base to check
distortion, aspect ratio, or node numbering by using
special reasoning.
***COPY SHAPE.KBS;1 INPUT.DAT;1
   .
   .
   .
??
You wish to look at the rules which would help you decide
if the model you have generated contains excessive element
distortion.
0.9
```

Figure 7-4--continued

```
**************************************
THE HYPOTHESIS THAT HAS CERTAINTY 0.900

For quadrilateral elements: keep interior corner angles
near 90 degree and always between 0 and 180 degrees.
For rectangular elements, the sides of the element should
be kept parallel to the coordinate axis in as far
possible.
For triangular elements: the element should be kept as
close as possible to an equilateral geometry.
If you think you followed these instruction and still have
distortion, then look for elements or nodes that you may
have inadvertently omitted.

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
You want to check the aspect ratio for the elements.
1.0

**************************************
THE HYPOTHESIS THAT HAS CERTAINTY 1.000

If displacements are the principal concern, then aspect
ratio should be less than 7.0.
If stresses are needed in addition, choose aspect ratio
less than 3.0.

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
You wish to look at the rules which would help you arrange
node numbers.
0.9

**************************************
THE HYPOTHESIS THAT HAS CERTAINTY 0.900

Smaller bandwidths obtained by numbering the nodes in the
direction of the smallest number of subdivision lines.
   .
   .
   .
```

Figure 7-4--continued

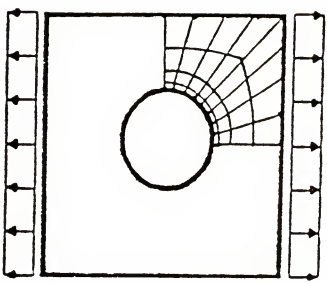


Figure 7-5. The finite element model for the plate problem.

Optimum Design Modeling

The third example is a cantilevered beam with a fixed tip mass shown in Figure 7-6. The structure is to be designed to minimize its weight and is subject to frequency constraints. An abbreviated map of an interactive session is shown in Figure 7-7.

The fourth example is an idealized swept wing structure shown in Figure 7-8. The structure is taken to be symmetric with respect to the x-y plane which corresponds to the wing middle surface. The upper half of the swept wing is modeled using sixty triangular membrane elements to represent the skin, seventy shear panel elements for the vertical webs, and twenty truss elements to represent forward and aft spar caps. This structure is designed to minimize weight and is subject to static loads condition as listed in Table 7-1. The application of the OPSYN expert system in the task of optimum design modeling results in the following system response, as shown in Figure 7-9.

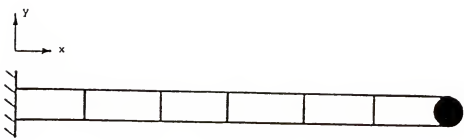


Figure 7-6. A cantilevered beam with a fixed tip mass.

```
$ COPY OPDEM.KBS;1 INPUT.DAT;1
$ RUN INFER
      .
      .
      .
  ??
  This problem is at the first step of optimum design
  modeling.
  1.0

  ************************************
  THE HYPOTHESIS THAT HAS CERTAINTY 1.000

  You need to consider if the computational time and/or
  accuracy are important for this problem.  You should run
  the APPMOD knowledge base by using backward-chaining
  reasoning
  ***COPY APPMOD.KBS;1 INPUT.DAT;3

  PLEASE INPUT 1 TO RUN THIS PROGRAM?
  1
$ RUN INFER
      .
      .
      .
  ??
  Reducing computational time is important for this problem.
  or The problem is expected to have more than 20 design
     variables or constraints.
  or The finite element model has more than 200 degrees of
     freedom.
  1.0
      .
      .
      .
  ??
  It is important to retain highly accurate analysis in this
  exercise.
  0.9

  ************************************
  THE HYPOTHESIS THAT HAS CERTAINTY 0.900

  The use of approximation concepts to reduce the
  dimensionality of the optimization problem is advised.
      .
      .
      .
$ RUN INFER
```

Figure 7-7. An interactive session of using OPDEM module for a cantilevered beam problem.

```
    .
    .
    .
??
Would you like to choose design variables for this
problem?
1.0

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
This problem has only one kind of structure.
1.0

**************************************
THE HYPOTHESIS THAT HAS CERTAINTY 1.000

You should run the VAR knowledge base to choose design
variables by using backward-chaining reasoning.
***COPY VAR.KBS;1 INPUT.DAT;3
    .
    .
    .
$ RUN INFER
    .
    .
    .
??
The geometric configuration of this problem is fixed.
1.0

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
This problem is a truss problem.
0.0

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
This problem deals with vibration analysis or is subjected
to dynamic loading.
1.0
```

Figure 7-7--continued

```
PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
The nonstructural mass in this problem is four to five
times larger than the structural mass.
0.95

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
This problem is a statically determinate problem.
0.95
   .
   .
   .
??
The cross-sectional dimensions are also subject to change.
1.0

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
This problem is one-dimensional beam structure.
1.0

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
The geometry of the structure is fixed.
1.0
   .
   .
   .
??
This problem is a beam or frame problem.
1.0

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
The cross-sectional properties (A, I, J) are fixed.
0.0
```

Figure 7-7--continued

```
**************************************
THE HYPOTHESIS THAT HAS CERTAINTY 0.902

Should choose the cross-sectional properties (A, I, J) as
design variables.  This would improve the linear
assumption for frequency constraints.
    .
    .
    .
$ RUN INFER
    .
    .
    .
??
Would you like to choose constraints for this problem?
1.0

**************************************
THE HYPOTHESIS THAT HAS CERTAINTY 1.000

You should run the CONST knowledge base to choose
constraints by using special reasoning.
***COPY CONST.KBA;1 INPUT.DAT;3
    .
    .
    .
$ RUN INFER
    .
    .
    .
**************************************
THE HYPOTHESIS THAT HAS CERTAINTY 1.000

Select natural vibration frequencies to formulate
inequality constraints.  The frequency can be larger than
lower bounds or less than upper bounds.  The constraint
is directly written in terms of the squares of the
circular frequency.
    .
    .
    .
??
This problem has geometrical restrictions on design
variables.
1.0
```

Figure 7-7--continued

```
**************************************
THE HYPOTHESIS THAT HAS CERTAINTY 1.000

In addition to the upper and lower bounds on design
variables (inequality constraints), you also need to add
the relationships between design variables as equality or
inequality constraints.
    .
    .
    .
$ RUN INFER
    .
    .
    .
??
You do not know which strategy is suitable for this
problem.
0.9

**************************************
THE HYPOTHESIS THAT HAS CERTAINTY 0.900

You should run the STRAT1 knowledge base to choose
suitable strategy by using backward-chaining reasoning.
***COPY STRAT1.KBS;1 INPUT.DAT;3
    .
    .
    .
$ RUN INFER
    .
    .
    .
??
The objective function and constraints are (or can be
represented in) a generalized positive polynomial form.
0.5

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
This problem is separable or can be approximated well as a
separable problem.  Separability exists when the objective
and constraints functions are each calculated as the sum
of functions of the individual design variables.
0.5
    .
    .
    .
```

Figure 7-7--continued

```
??
This problem does not use any approximation concepts.
0.0

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
This problem has some stress constraints.
0.2

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
This problem does not have any stress constraints.
0.9

**************************************
THE HYPOTHESIS THAT HAS CERTAINTY 0.900

Choose mathematical programming method and approximation
concepts for this problem.
   .
   .
   .
```

Figure 7-7--continued

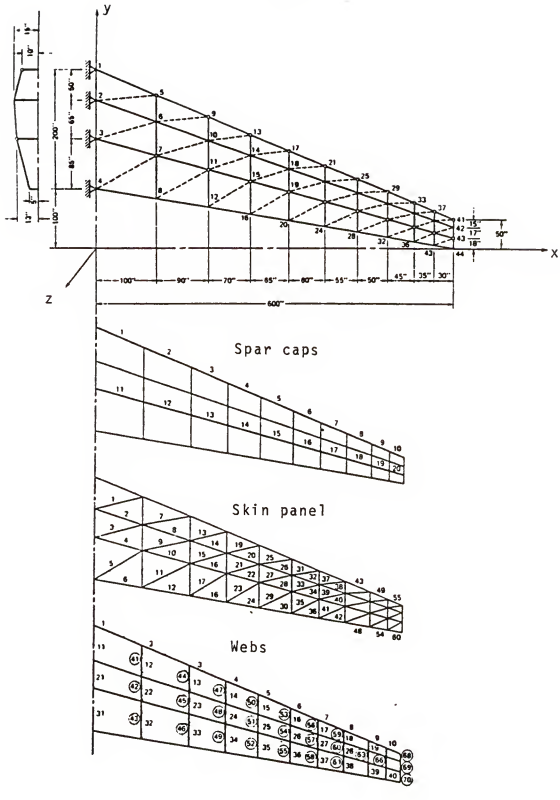


Figure 7-8. Finite element model for swept wing.

Table 7-1. Load condition data for swept wing.

For all nodes, $P_x = 0.0$ and $P_y = 0.0$

Node No.	P_z (lbs)	Node No.	P_z (lbs)	Node No.	P_z (lbs)
Load Condition 1					
5	1282.0	19	1453.0	33	206.0
6	2581.0	20	1057.0	34	431.0
7	3398.0	21	459.0	35	563.0
8	2380.0	22	958.0	36	383.0
9	978.0	23	1251.0	37	144.0
10	2013.0	24	852.0	38	302.0
11	2593.0	25	362.0	39	395.0
12	1764.0	26	756.0	40	269.0
13	727.0	27	986.0	41	62.0
14	1386.0	28	671.0	42	129.0
15	1906.0	29	282.0	43	169.0
16	1297.0	30	589.0	44	116.0
17	570.0	31	768.0		
18	1190.0	32	522.0		

```
$ COPY OPDEM.KBS;1 INPUT.DAT;1
$ RUN INFER
     .
     .
     .
  PLEASE ANSWER THE FOLLOWING QUESTION WITH A
  FACTOR RANGING FROM
  0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
  ??
  This problem is at the first step of optimum design
  modeling.
  1.0

  ************************************
  THE HYPOTHESIS THAT HAS CERTAINTY 1.000

  You need to consider if the computational time and/or
  accuracy are important for this problem.  You should run
  the APPMOD knowledge base by using backward-chaining
  reasoning.
  ***COPY APPMOD.KBS;1 INPUT.DAT;3

  PLEASE INPUT 1 TO RUN THIS PROGRAM?
  1
$ RUN INFER

  PLEASE ANSWER THE FOLLOWING QUESTION WITH A
  FACTOR RANGING FROM
  0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
  ??
  Reducing computational time is important for this problem.
  or The problem is expected to have more than 20 design
  variables or constraints.
  or The finite element model has more than 200 degrees of
  freedom.
  1.0
     .
     .
     .
  ??
  It is important to retain highly accurate analysis in this
  exercise.
  1.0

  ************************************
  THE HYPOTHESIS THAT HAS CERTAINTY 1.000

  The use of approximation concepts to reduce the
  dimensionality of the optimization problem is advised.
```

Figure 7-9. An interactive session of using the OPDEM module for a swept wing structure problem.

```
    .
    .
    .
$ RUN INFER
    .
    .
    .
 ??
 Would you like to choose design variables for this
 problem?
 0.9

 PLEASE ANSWER THE FOLLOWING QUESTION WITH A
 FACTOR RANGING FROM
 0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
 ??
 This problem has only one kind of structure.
 0.0

 PLEASE ANSWER THE FOLLOWING QUESTION WITH A
 FACTOR RANGING FROM
 0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
 ??
 This problem has more than one kind of structure.
 1.0

 ****************************************
 THE HYPOTHESIS THAT HAS CERTAINTY 0.900

 You should run the VAR knowledge base to choose design
 variables by using special reasoning.
 ***COPY VAR.KBS;1 INPUT.DAT;3
    .
    .
    .
 ??
 The geometric configuration of this problem is fixed.
 1.0

 PLEASE ANSWER THE FOLLOWING QUESTION WITH A
 FACTOR RANGING FROM
 0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
 ??
 This problem is a truss problem.
 0.95
    .
    .
    .
 ??
 This problem is a statically indeterminate problem.
 1.0
```

Figure 7-9--continued

.
.
.
??
There are two to four load paths available.
0.95

****
THE HYPOTHESIS THAT HAS CERTAINTY 0.902

This problem can have the inverse of cross-sectional area (1/A) as design variables. This would improve the linear approximation of the stress and displacement constraints.
.
.
.
??
The structure contains shear panels or membrane elements.
0.95

****
THE HYPOTHESIS THAT HAS CERTAINTY 0.950

Choose the thickness (t) of shear panels or membrane elements as design variables.
.
.
.
??
The structural members of this problem consider yielding failure under tension or compression loads.
0.9

****
THE HYPOTHESIS THAT HAS CERTAINTY 0.900

Choose stresses to formulate inequality constraints. The Stresses in each member should be less than prescribed allowable values.
.
.
.

Figure 7-9--continued

Optimization Performance

The last example is a space frame idealization of a helicopter tail-boom structure subject to a single loading condition as shown in Figure 7-10. All of the members have the same cross sectional shape. The structure is designed for minimum mass subject to stress and displacement constraints. The design variables are the sizing variables, mean radius R, and thickness t, in each member. An abbreviated map of an interactive session is listed in Figure 7-11.

The designer, based on the information available in the explanation, can choose a hypothesis that has the highest confidence level associated with it as the optimum choice. For the above problem, the suggested choice is a one-dimensional-search of 8 which corresponds to the method of using polynomial interpolation/extrapolation without first finding bounds.

P_1 = 1.4903 kips; P_2 = 1.6918 kips
P_3 = 0.1400 kips; P_4 = 1.3658 kips

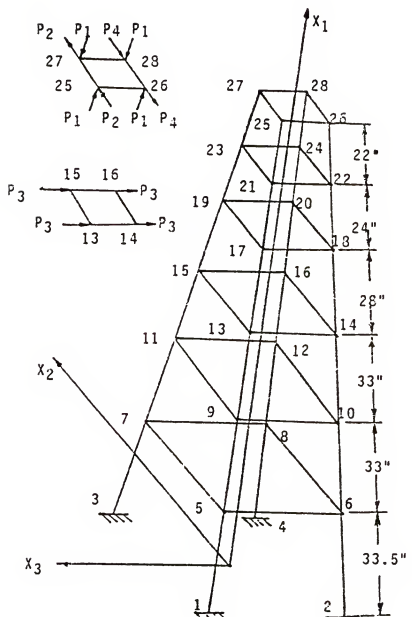


Figure 7-10. A space frame idealization of a helicopter tail-boom structure.

```
$ COPY OPPER.KBS;1 INPUT.DAT;1
$ RUN INFER
    .
    .
    .
??
This problem is at the first step of selecting an
optimization algorithm and has not been performed by any
algorithms.
1.0

**************************************
THE HYPOTHESIS THAT HAS CERTAINTY 1.000
You need to consider if the efficiency and/or reliability
are important for this problem.  You should run the METHOD
knowledge base by using backward-chaining reasoning.
***COPY METHOD.KBS;1 INPUT.DAT;3
    .
    .
    .
??
Efficiency (reducing the computational time) is more
important than reliability for this problem.
0.9

**************************************
THE HYPOTHESIS THAT HAS CERTAINTY 0.900
Treatment of the problem as a constrained problem is
advised.
    .
    .
    .
??
You do not know which strategy is suitable for this
problem.
1.0

**************************************
THE HYPOTHESIS THAT HAS CERTAINTY 0.900
You should run the CSTRAT knowledge base to choose a
strategy by using backward-chaining reasoning.
***COPY CSTRAT.KBS;1 INPUT.DAT;3
    .
    .
    .
??
The objective function and/or constraints are nonlinear
for this problem.
0.9
```

Figure 7-11. An interactive session of using OPPER module for a helicopter tail-boom structure problem.

```
PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
Function evaluations are expensive for this problem.
1.0

**************************************
THE HYPOTHESIS THAT HAS CERTAINTY 0.900
You should choose strategy of 0 for this problem.
Strategy 0 means no strategy is to be used.
   .
   .
   .
**************************************
THE HYPOTHESIS THAT HAS CERTAINTY 0.900
You should choose optimizer of 4 for this problem.
Optimizer 4 is the method of feasible directions.
   .
   .
   .
??
The objective function and/or constraints are nonlinear
for this problem.
1.0

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
This problem has a well-conditioned objective function
and/or constraints. (A problem with a smoothly varying
objective function, constraints, and derivatives e.g.
polynomials.)
0.7

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
This problem has an objective function and/or constraints
which are nearly linear or quadratic.
0.6
   .
   .
   .
??
There are less active constraints than design variables
expected at the optimum for this problem.
0.8
```

Figure 7-11--continued

```
PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
This problem requires a strategy of 0.
1.0
   .
   .
   .
??
This problem is a convex problem.
0.2

PLEASE ANSWER THE FOLLOWING QUESTION WITH A
FACTOR RANGING FROM
0.0 (DEFINITE "NO") TO 1.0 (DEFINITE "YES")
??
Relative minima are likely to exist for this problem.
0.8

          ***NO CONCLUSION FOR THIS PROBLEM***

OPTIONS ARE
 1 --> SEE AN EXPLANATION
 2 --> CONTINUE TO NEXT CASE
 3 --> EXIT
? PLEASE ANSWER WITH THE NUMBER !
1
   .
   .
   .
SO THE FOLLOWING HYPOTHESIS HAS 0.805 CERTAINTY -->
You should choose one-dimensional-search of 8 for this
problem.  A one-dimensional-search of 8 uses polynomial
interpolation/extrapolation without first finding bounds.
   .
   .
   .
SO THE FOLLOWING HYPOTHESIS HAS 0.48 CERTAINTY -->
You should choose one-dimensional-search of 7 for this
problem.  A one-dimensional-search of 7 first finds bounds
then uses polynomial interpolation.
   .
   .
   .
FORTRAN STOP
```

Figure 7-11--continued

CHAPTER VIII

CONCLUDING REMARKS

This study brings the techniques of artificial intelligence into the optimum structural design domain. A knowledge-based expert system approach for optimum structural synthesis is presented. This approach results in the accumulation of a body of heuristic knowledge about the optimum structural design domain into a computer and simulates the human expert's reasoning processing to provide knowledgeable advice about the intricate and involved task of optimum structural synthesis.

A prototype expert system, OPSYN, has been developed to aid a structural designer in the computer-aided optimum design of structural and mechanical systems. The OPSYN expert system can provide interactive assistance in finite element modeling for finite element program EAL, optimum structural design modeling, and selection of optimization strategies and parameters for the general-purpose optimization programs ADS.

The OPSYN represents knowledge in the IF-THEN rule-based scheme and separates the knowledge base into different knowledge modules for using suitable inference-reasoning techniques and speeding up the inference process. Currently, there are about 264 rules within the knowledge

base. The graphical display capability within the CAD environment was used to represent some engineering knowledge more concisely than conventional text-based representation.

The environment for the OPSYN expert system development is the INFER inference engine. INFER was especially developed for the building of expert systems for engineering design and analysis and was intended to be operational in a CAD environment. It is written in FORTRAN-77 which facilitates understanding for most structural engineers. INFER allows the user to associate confidence levels with their responses and to choose appropriate inference-reasoning techniques. The available techniques are forward-chaining reasoning, backward-chaining reasoning, and special reasoning. It also provides an explanation facility, a graphical display facility, and some special functions to communicate with conventional algorithmic programs for engineering analysis.

During the OPSYN expert system development, the knowledge-acquisition process was recognized as the most difficult task. An approach of using existing CAD capabilities as a tool to elicit knowledge from the domain expert directly was presented. This approach requires several domain experts to perform the tasks on a set of prestructured problems. The responses from domain experts are stored in a file, which can be analyzed to propose rules for the knowledge base. A prototype automated knowledge-acquisition system for finite element modeling, embedded in

a CAD environment, was developed in this study. A knowledge-base editor facility was also developed to aid the knowledge engineer in editing and modifying the knowledge base.

The preliminary testing and verification of the current approach used in the OPSYN expert system gives satisfactory results. It is believed that a system like OPSYN would enhance the role of structural optimization in the design community.

However, a significant limitation of the current OPSYN expert system is that it operates only on the basis of responses provided by the user. Additional development work is necessary on the subject of utilizing the existing capability of communicating with conventional analysis programs in the INFER environment. This can be used to obtain more reliable and quantitative information about the problem domain.

The task in this direction is to use these general rules in the FEMOD module to derive macros that have become routine in automated finite element analysis (interactive graphics-based automated mesh generation [14] or automated input file generator [78]). For a complete automated finite element analysis, the work can be extended to automate interpretation of the finite element analysis results. The use of symbolic manipulations [79] in the development of an optimum design model can be combined with the OPDEM module to automate the optimum design modeling process in the

future. By utilizing the capability of dealing with external programs in the INFER environment, the OPPER module can be extended to establish program monitoring and diagnostic capabilities, such as recognition of numerical errors, identification of crucial design variables and constraints, and existence of ill-defined design spaces. This would allow structural optimization to become a friendly and reliable structural design tool.

As for the issue of knowledge representation, the current structure of knowledge base and the scheme for knowledge representation is adequate for this initial concept demonstration of the expert system. The use of graphical displays is particularly attractive to prevent misrepresentation of knowledge. Use of the graphical display capability to implement the knowledge representation will become more important in the engineering expert systems, since this form of representation plays an important role in engineering practice. The use of other forms of knowledge representation can be considered in future studies as the OPSYN expert system adapts to other tasks and knowledge with more complex relationships.

Since OPSYN is a concept demonstration expert system, the knowledge contained in the knowledge base is only a small portion of the total knowledge. There is a need to elicit more knowledge from domain experts to extend the capability of knowledge base. The task of eliciting the

knowledge from domain experts efficiently will be become a major issue in all current knowledge-engineering research.

Before automated machine-learning techniques [80] attain a more mature status, the knowledge-acquisition process will continue to be a bottleneck in the development of the expert system. To improve the current knowledge-acquisition system, there is a need to consider other techniques such as the general induction method and the repertory grid method, to induct rules from the examples.

Finally, the goal of building truly intelligent CAD systems may be realized by integrating expert systems and automated knowledge-acquisition methods into a CAD environment. However, there is a substantial amount of research that must be conducted before this goal can be realized.

APPENDIX A

A DESCRIPTION OF ELEMENTS IN EAL

The EAL (Engineering Analysis Language) [74] is a finite element computer program. Its primary applications are engineering analysis and design based on structural and thermal finite element methods. A brief description of the available structural elements in EAL program is presented next.

The available one-dimensional (two-node) elements are as follows: (a) E21 are general straight or circularly curved beam elements such as channels, wide-flanges, angles, tubes, zees, etc; (b) E22 is a beam element for which the user wishes to directly specify the 6 by 6 intrinsic stiffness matrix; (c) E23 is a simple axial (truss) element which supports only axial force; (d) E24 is a planar beam element with axial stiffness, and for which there is only one plane of bending stiffness, but no torsional stiffness, and (e) E25 is a zero-length (spring) element used to elastically connect geometrically coincident joints.

The EAL also contains the following two-dimensional elements: (a) E31 and E41 are triangular and quadrilateral membrane elements, respectively; (b) E32 and E42 are triangular and quadrilateral plate bending elements, respectively; (c) E33 and E43 are triangular and

quadrilateral combined membrane and bending elements, respectively, and (d) E44 is a shear panel element which is essentially identical to the E41 elements, except the assumed stress field consists of pure shear. All of the above elements may have anisotropic constitutive relations.

The available three-dimensional elements in EAL are as follows: (a) S41 is a four-node pyramid (tetrahedron) element; (b) S61 is a six-node wedge (pentahedron) element, and (c) S81 is an eight-node brick (hexahedron) element. Stress-strain relations of the above three-dimensional elements may be anisotropic.

APPENDIX B

ADS: A GENERAL-PURPOSE OPTIMIZATION PROGRAM

The ADS (Automated Design Synthesis) version 1.10 computer program [75] is a general-purpose numerical optimization program containing a wide variety of algorithms. A brief description of the available options in the ADS program is provided next.

The procedure of solving an optimization problem in the ADS program is separated into three levels which are identified as strategy, optimizer, and one-dimensional search. The task in the strategy level is to convert the original optimization problem to one which can be solved more easily with the existing methods. The actual optimization is performed at the optimizer level which determines the desired search direction. Finally, the one-dimensional search level performs a line search to minimize the objective function in a direction specified by the optimizer level. At each level, there are several options available. Tables B-1 to B-3 list the available strategies, optimizers, and one-dimensional search options, respectively. Table B-4 identifies the combinations of algorithms which are available in the ADS program.

Table B-1. Strategy options.

ISTRAT	Strategy to be used
0	None. Go directly to the optimizer.
1	Sequential unconstrained minimization using the exterior penalty function method.
2	Sequential unconstrained minimization using the linear extended interior penalty function method.
3	Sequential unconstrained minimization using the quadratic extended interior penalty function method.
4	Sequential unconstrained minimization using the cubic extended interior penalty function method.
5	Augmented Lagrange Multiplier method.
6	Sequential Linear Programming.
7	Method of Centers (method of inscribed hyperspheres).
8	Sequential Quadratic Programming.
9	Sequential Convex Programming.

Table B-2. Optimizer options.

IOPT	Optimizer to be used
1	Fletcher-Reeves algorithm for unconstrained minimization.
2	Davidon-Fletcher-Powell (DEP) variable metric method for unconstrained minimization.
3	Broydon-Fletcher-Goldfarb-Shanno (BFGS) variable metric method for unconstrained minimization.
4	Method of Feasible Directions (MFD) for constrained minimization.
5	Modified Method of Feasible Directions for constrained minimization.

Table B-3. One-dimensional search options.

IONED	One-dimensional search option
1	Find the minimum of an unconstrained function using the Golden Section method.
2	Find the minimum of an unconstrained function using the Golden Section method followed by polynomial interpolation.
3	Find the minimum of an unconstrained function by first finding bounds and then using polynomial interpolation.
4	Find the minimum of an unconstrained function by polynomial interpolation/extrapolation without first finding bounds on the solution.
5	Find the minimum of a constrained function using the Golden Section method.
6	Find the minimum of a constrained function using the Golden Section method followed by polynomial interpolation.
7	Find the minimum of a constrained function by first finding bounds and then using polynomial interpolation.
8	Find the minimum of a constrained function by polynomial interpolation/extrapolation without first finding bounds on the solution.

Table B-4. Program options.

Strategy	Optimizer 1	2	3	4	5
0	X	X	X	X	X
1	X	X	X	0	0
2	X	X	X	0	0
3	X	X	X	0	0
4	X	X	X	0	0
5	X	X	X	0	0
6	0	0	0	X	X
7	0	0	0	X	X
8	0	0	0	X	X
9	0	0	0	X	X
One-dimensional serach					
1	X	X	X	0	0
2	X	X	X	0	0
3	X	X	X	0	0
4	X	X	X	0	0
5	0	0	0	X	X
6	0	0	0	X	X
7	0	0	0	X	X
8	0	0	0	X	X

Note: An X denotes an allowed combination of algorithms.

REFERENCES

1. Noor, A. K., and Pilkey, W. D. (eds.), State-of-the-Art Surveys on Finite Element Technology, ASME, New York, 1983.

2. Zienkiewicz, O. C., The Finite Element Method, third edition, McGraw-Hill, New York, 1977.

3. Fox, R. L., Optimization Methods for Engineering Design, Addison-Wesley Publishing Company, Reading, MA., 1971.

4. Vanderplaats, G. N., Numerical Optimization Techniques for Engineering Design with Application, McGraw-Hill, New York, 1984.

5. Hajela, P., "Structural Optimization Methodologies in the CAD Environment," paper presented at the 22nd SES Annual Technical Meeting, ESP22/85048, Oct. 7-9, 1985.

6. Gero, J. S. (ed.), Optimization in Computer-Aided Design, Elsevier Science Publisher B. V., Amsterdam, The Netherlands, 1985.

7. Esping, B. J. D., and Holm, D., "A CAD Approach to Structural Optimization," Proceedings of NATO/NASA/NSF/USAF Advance Study Institute on Computer-Aided Optimal Design, Vol. 3 (ed. Mota Soares), Troia, Portugal, 1986, pp. 201-214.

8. Wallerstein, D. V., "Design Enhancement Tools in MSC/NASTRAN," NASA CP-2327, Part I, NASA, Washington, D. C., 1984.

9. Rouse, N. E., "Design Optimization Goes Commercial," Machine Design, Vol. 58, No. 25, Oct. 23, 1986, pp. 84-87.

10. Bennett, J. A., and Botkin, M. E., "Automated Design for Automotive Structures," ASME Journal of Mechanical Design, Vol. 104, Oct. 1982, pp. 799-805.

11. Brama, T., "Weight Optimization of Aircraft Structures," Proceedings of NATO/NASA/NSF/USAF Advance Study Institute on Computer-Aided Optimal Design, Vol. 3 (ed. Mota Soares), Troia, Portugal, 1986, pp. 267-277.

12. Schmit, L. A., and Miura, H., "Approximation Concepts for Efficient Structural Synthesis," NASA CR-2552, NASA, Washington, D. C., 1976.

13. Vanderplaats, G. N., Miura, H., and Chargin, M., "Large Scale Structural Synthesis," Finite Elements in Analysis and Design, Vol. 1, No. 3, 1985, pp. 117-130.

14. Krouse, J. K., "Reducing the Labor Content in Finite-Element Analysis," Machine Design, Vol. 55, No. 20, Sept. 8, 1983, pp. 64-69.

15. Smith, G. E., "The Dangers of CAD," Mechanical Engineering, Vol. 108, No. 2, Feb. 1986, pp. 58-64.

16. Dym, C. L., "EXPERT SYSTEMS: New Approaches to Computer-Aided Engineering," Engineering with Computers, Vol. 1, No. 1, 1985, pp. 9-25.

17. Winston, P. H., Artificial Intelligence, Addison-Wesley Publishing Company, Reading, MA., 1984.

18. Woodward, W. S., and Morris, J. W., "Improving Productivity in Finite Element Analysis through Interactive Processing," Finite Elements in Analysis and Design, Vol. 1, No. 1, 1985, pp. 35-48.

19. Fenves, S. J., "A Framework for a Knowledge-Based Finite Element Assistant," Applications of Knowledge-Based Systems to Engineering Analysis and Design (ed. C. L. Dym), ASME, New York, 1985, pp. 1-7.

20. Shephard, M. S., "Finite Element Modeling Within an Integrated Geometric Modeling Environment, Part II: Attribute Specification, Domain Differences, and Indirect Element Types," Engineering with Computers, Vol.1, No. 2, 1985, pp. 73-85.

21. Bennett, J., Creart, L., Englemore, R., and Melosh, R., "SACON: A Knowledge-Based Consultant for Structural Analysis," Technical Report STAN-CS-78-699, Stanford University, Stanford, California, 1978.

22. Rivlin, J. M., Hsu, M. B., and Marcal, P. V., "Knowledge-Based Consultation for Finite Element Structural Analysis," U. S. Air Force Flight Dynamics Laboratory Report AFWAL-TR-80-3069, Wright-Patterson Air Force Base, Dayton, Ohio, 1980.

23. Lu, S. C.-Y., "A Consultative Expert System for Finite Element Modeling of Strip Drawing," paper presented at the XIII North American Manufacturing Research Conference, University of California at Berkeley, May 1985.

24. Holt, R. H., and Narayana, U. V. L., "Adding Intelligence to Finite Element Modeling," Expert Systems in Government Symposium (eds. K. N. Karna, K. Parsaye, and B. G. Silverman), IEEE, New York, 1986, pp. 326-337.

25. Arora, J. S., and Baenziger, G., "Use of AI in Design Optimization," Computer Methods in Applied Mechanics and Engineering, Vol. 54, No. 3, 1986, pp. 303-323.

26. Papalambros, P., "Knowledge-Based Systems in Optimal Design," Proceedings of NATO/NASA/NSF/USAF Advance Study Institute on Computer-Aided Optimum Design, Vol. 3 (ed. Mota Soares), Troia, Portugal, 1986, pp. 311-362.

27. Jha, N. K., "Engineering Optimization and Expert Systems," Proceedings of the 1985 ASME International Computers in Engineering Conference, Vol. 3, ASME, New York, 1985, pp. 97-101.

28. Jozwiak, S. F., "Application of Artificial Intelligence Notions in Structural Optimization Programs," Computers and Structures, Vol. 24, No. 6, 1986, pp. 1009-1013.

29. Li, H. L., and Papalambros, P., "A Production System for Use of Global Optimization Knowledge," ASME Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 107, No. 2, June 1985, pp. 277-284.

30. Rogers, J. L., Jr., and Barthelemy, Jean-Francois, M., "An Expert System for Choosing the Best Combination of Options in a General Purpose Program for Automated Design Synthesis," Engineering with Computers, Vol. 1, No. 4, 1985, pp. 217-227.

31. Vanderplaats, G. N., Sugimoto, H., and Sprague, C. M., "ADS-1: A New General-Purpose Optimization Program," AIAA Journal, Vol. 22, No. 10, Oct. 1984, pp. 1458-1459.

32. Baenziger, G., and Arora, J. S., "Development of an Artificial Intelligent Nonlinear Optimization Expert System," Proceedings of the 27th AIAA/ASME/ASCE/AHS Structures, Structural Dynamics and Materials Conference, Part II, AIAA, New York, 1986, pp. 67-77.

33. Hartmann, D., "Application of AI-Tools for Re-Analysis within Structural Optimization," Proceedings of NATO/NASA/NSF/USAF Advance Study Institute on Computer-Aided Optimal Design, Vol. 3 (ed. Mota Soares), Troia, Portugal, 1986, pp. 415-430.

34. Maxwell, C., Scientific Papers, Vol. 2, 1869, Dover Publications, New York, 1952, pp. 175-177.

35. Michell, A. G. M., "The Limits of Economy of Material in Frame Structures," Philosophical Magazine, Series 6, Vol. 8, 1904.

36. Khot, N. S., Venkayya, V. B., and Berke, L., "Comparison of Optimality Criteria Algorithms for Minimum Weight Design of Structures," AIAA Journal, Vol. 17, 1979, pp. 182-190.

37. Schmit, L. A., "Structural Design by Systematic Synthesis," Proceedings of the 2nd Conference on Electronic Computation, American Society of Civil Engineering, New York, 1960, pp. 105-122.

38. Vanderplaats, G. N., "Structural Optimization--Past, Present, and Future," AIAA Journal, Vol. 20, No. 7, July 1982, pp. 992-1000.

39. Haftka, R. T., and Kamat, M. P., Elements of Structural Optimization, Martinus Nijhoff Publishers, Boston, MA., 1985.

40. Mair, W. M., "The Objectives of the National Agency for Finite Element Methods and Standards," Computers and Structures, Vol. 21, No. 5, 1985, pp. 875-879.

41. Macneal, R. H., and Harder, R. L., "A Proposed Standard Set of Problems to Test Finite Element Accuracy," Finite Elements in Analysis and Design, Vol. 1, No. 1, April 1985, pp. 3-20.

42. Kikuchi, N., "Adaptive Grid-Design Methods for Finite Element Analysis," Computer Methods in Applied Mechanics and Engineering, Vol. 55, Nos. 1-2, 1986, pp. 129-160.

43. Sobieszczanski-Sobieski, J., "A Linear Decomposition Method for Larger Optimization Problems--Blueprint for Development," NASA TM-83248, NASA, Washington, D. C., 1982.

44. Hajela, P., and Sobieszczanski-Sobieski, J., "The Controlled Growth Method--A Tool for Structural Optimization," AIAA Journal, Vol. 20, No. 10, Oct. 1982, pp. 1440-1441.

45. Prasad, B., "Novel Concepts for Constraint Treatments and Approximations in Efficient Structural Synthesis," AIAA Journal, Vol. 22, No. 7, July 1984, pp. 957-966.

46. Kirsch, U., Optimum Structural Design-Concepts, Methods and Applications, Mcgraw-Hill, New York, 1981.

47. Hajela, P., and Chen, J. L., "Optimum Structural Sizing of Conventional Cantilever and Joined Wing Configurations Using Equivalent Beam Models," AIAA-86-2653, paper presented at the AIAA/AHS/ASEE Aircraft Systems, Design & Technology Meeting, Dayton, Ohio, Oct. 20-22, 1986.

48. Gallagher, R. H., "Fully Stressed Design," Optimum Structural Design, Theory and Applications, (eds. R. H. Gallagher and O. C. Zienkiewicz), John Wiley & Sons Ltd., New York, 1973, pp. 19-32.

49. Hajela, P., "Geometric Programming Strategies in Large Scale Structural Synthesis," AIAA Journal, Vol. 24, No. 7, July 1986, pp. 1173-1178.

50. Fleury, C., and Schmit, L. A., "Dual Methods and Approximation Concepts in Structural Synthesis," NASA CR-3226, NASA, Washington, D. C., 1980.

51. Haftka, R. T., "Sensitivity Calculations for Iteratively Solved Problems," International Journal for Numerical Methods in Engineering, Vol. 21, 1985, pp. 1535-1546.

52. Arora, J. S., and Haug, E. J., "Methods of Design Sensitivity Analysis in Structural Optimization," AIAA Journal, Vol. 17, No. 9, Sept. 1979, pp. 970-974.

53. Belegundu, A. D., and Arora, J. S., "A Study of Mathematical Programming Methods for Structural Optimization, Part I: Theory; Part II: Numerical Aspects," International Journal for Numerical Methods in Engineering, Vol. 21, 1985, pp. 1583-1623.

54. Sandgren, E., and Ragsdell, K. M., "The Utility of Nonlinear Programming Algorithms: A Comparative Study-parts I and II," ASME Journal of Mechanical Design, Vol. 102, July 1980, pp. 540-552.

55. Carpenter, W. C., and Smith, E. A., "Computational Efficiency of Nonlinear Programming Methods on a Class of Structural Problems," International Journal for Numerical Methods in Engineering, Vol. 11, 1977, pp. 1203-1223.

56. Rajan, S. D., "A Hybrid Nonlinear Programming Method for Design Optimization," Journal of Structural Mechanics, Vol. 14, No. 4, 1986, pp. 455-474.

57. Waterman, D. A., A Guide to Expert Systems, Addison-Wesley Publishing Company, Reading, MA., 1986.

58. Hayes-Roth, F., Waterman, D. A., and Lenat, D. B. (eds.), Building Expert Systems, Addison-Wesley Publishing Company, Reading, MA., 1983.

59. Rice, E., Artificial Intelligence, McGraw-Hill, New York, 1983.

60. Davis, R., "Applications of Meta Level Knowledge to the Construction, Maintenance and Use of Large Knowledge Bases," Report STAN-CS-76-552, Stanford AI Laboratory, Stanford University, Stanford, California, 1976.

61. Genesereth, M. R., and Ginsberg, M. L., "Logic Programming," Communications of the ACM, Vol. 28, No. 9, Sept. 1985, pp. 933-941.

62. Chang, C. L., and Lee, R. C., Symbolic Logic and Mathematical Theorem Proving, Academic Press, New York, 1973.

63. Clocksin, W. F., and Mellish, C. S., Programming in PROLOG, Springer-Verlag, Heidelberg, 1981.

64. Hart, A., "Knowledge Elicitation: Issues and Methods," Computer-Aided Design, Vol. 17, No. 9, Nov. 1985, pp. 455-462.

65. Rada, R., "Automating Knowledge Acquisition," EXPERT SYSTEMS, Principles and Cases Studies, (ed. R. Forsyth), Chapman and Hall, New York, 1984, pp. 190-210.

66. Hart, A., "The Role of Induction in Knowledge Elicitation," Expert Systems, Vol. 2, No. 1, Jan. 1985, pp. 24-28.

67. Michalski, R. S., "Pattern Recognition as Rule-Guided Inductive Inference," IEEE Transactions of Pattern Analysis and Machine Intelligence, Vol. 2, No. 4, 1980. pp. 349-361.

68. Kelly, G. A., The Psychology of Personal Constructs, Norton, New York, 1955.

69. Boose, J. H., "A Knowledge Acquisition Program for Expert Systems Based on Personal Construct Psychology," International Journal of Man-Machine Studies, Vol. 23, 1985, pp. 495-525.

70. Barber, G. R., "LISP vs. C for Implementing Expert Systems," AI EXPERT, Vol. 2, No. 2, Feb. 1987, pp. 28-31.

71. Kornell, J., "Embedded Knowledge Acquisition to Simplify Expert Systems Development," Applied Artificial Intelligence Reporter, Aug.-Sept. 1984, pp. 28-30.

72. Riese, C., "Control Strategies in RULEMASTER," Report R1-RS-00296, Radian Corporation, Austin, Tex., 1985.

73. Gilmore, J. F. and Pulaski, K., "A Survey of Expert System Tools," IEEE Proceedings of the 2nd Conference on Artificial Intelligence Applications, IEEE Computer Society, New York, 1985, pp. 498-502.

74. Whetstone, D., EISI-EAL Engineering Analysis Language Reference Manual--EISI-EAL System Level 2091, Engineering Information Systems, Inc., San Jose, California, July 1983.

75. Vanderplaats, G. N., "ADS--A FORTRAN Program for Automated Design Synthesis--Version 1.10," NASA CR-177985, NASA, Washington, D. C., 1985.

76. van Melle, W., Shortliffe, E. H., and Buchanan, B. G., "EMYCIN: A Knowledge Engineer's Tool for Constructing Rule-Based Expert Systems," Rule-Based Expert Systems, (eds. B. Buchanan and E. Shortliffe), Addison-Wesley Publishing Company, Reading, MA., 1984, pp. 302-328.

77. Chen, J. L., and Hajela, P., "OPSYN: A Consultative Expert System for Optimum Structural Synthesis," Department of Engineering Sciences Report ENS-87-8-1, University of Florida, Gainesville, Florida, August 1987.

78. McDonald, P. D., "A Program Generator for Producing Finite Element Model Generator Software," Finite Element Methods, Modeling, and New Applications, (eds. E. M. Patton, H. Chung, F. Hatt, D. Hui, and H. A. Kamel), ASME, New York, 1986, pp. 27-32.

79. Krishnaswami, P. and Bhatti, M. A., "Symbolic Computing in Optimal Design of Dynamic Systems," ASME Paper No. 85-DET-76, ASME, New York, 1985.

80. Michalaki, R., Carbonell, J., and Mitchell, T. (eds.), Machine Learning, Tioga Publishing, Palo Alto, California, 1983.

BIOGRAPHICAL SKETCH

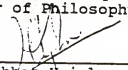
Jahau Lewis Chen was born September 1, 1956, in Tainan, Taiwan, Republic of China. He received a bachelor's degree in naval architecture and marine engineering from National Cheng Kung University, Taiwan, in June 1978.

In August, 1982, he enrolled in the graduate school at Old Dominion University in Norfolk, Virginia, and received a Master of Engineering degree in mechanical engineering and mechanics in August 1984. Currently, he is pursuing the Doctor of Philosophy degree in engineering mechanics at the University of Florida.

His research interests are in the area of expert systems and optimal design.

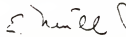
He is a member of Tau Beta Pi and Phi Kappa Phi.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Dr. Prabhat Hajela, Chairman
Assistant Professor of
Engineering Sciences

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Dr. Gale E. Nevill, Jr.
Professor of Engineering
Sciences

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Dr. Martin A. Eisenberg
Professor of Engineering
Sciences

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Dr. Chen-Chi Hsu
Professor of Engineering
Sciences

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Dr. Marc I. Hoit
Assistant Professor of Civil Engineering

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

August 1987

Herbert A. Bevis
Dean, College of Engineering

Dean, Graduate School